

Underwater Vehicles

Modeling,

Duc Cuong Vu

*Motion Control and Applied Robotics Laboratory,
School of Electrical and Electronic Engineering,
Hanoi University of Science and Technology*

Version: 1.0

Last updated: March 17, 2025



Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



Introduction

Handbooks and references

UUV - Unmanned Underwater vehicle

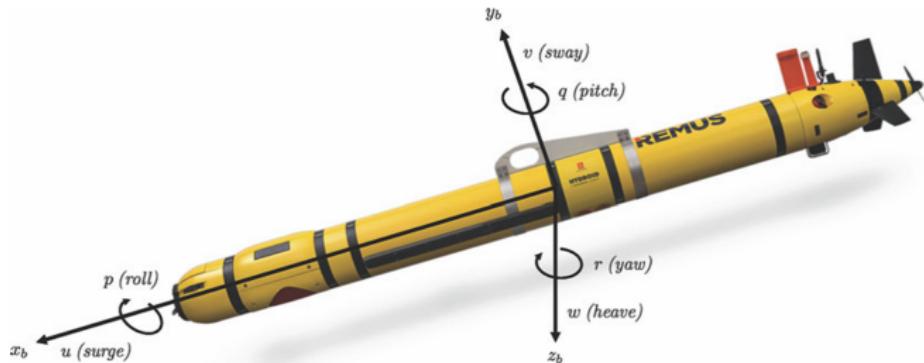


Figure: 6-DOF motion variables for a UUV/AUV

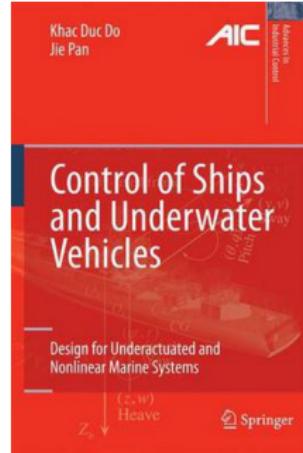
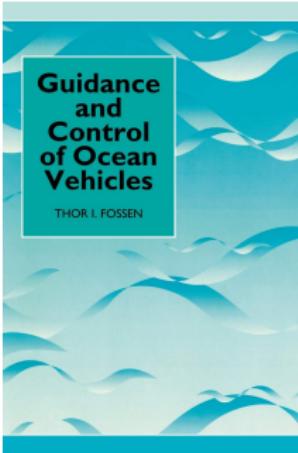
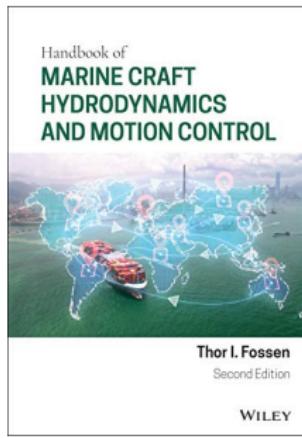
Handbooks & References:

- Thor I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, 2011.
- Thor I. Fossen, *Guidance and Control of Ocean Vehicles*, Wiley, 1994.



Introduction

Handbooks and references



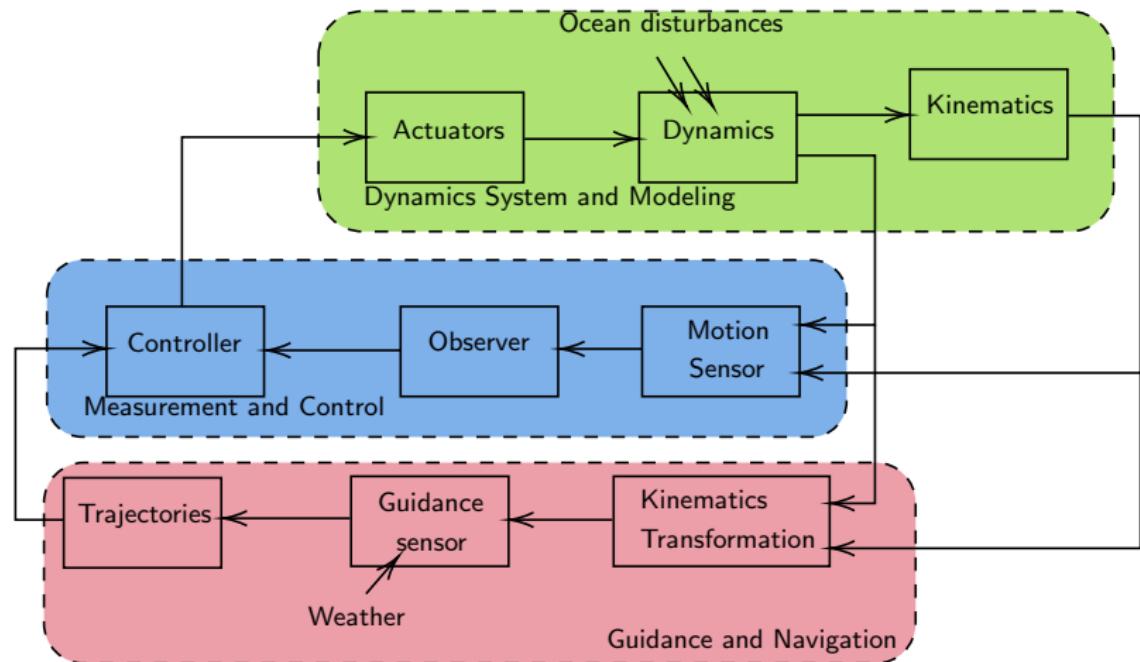
Handbooks & References:

- Review articles, Research articles, Handbook, ...
 - IEEE Explore, Springer, Elservier, Wiley, Taylor & Francis, ...



Introduction

Automatic Weather Routing



Introduction

Model classifications

Degree of Freedom classifications under purpose

- 1 DOF → model can be used to design the forward speed controller and heading autopilots and damping system
- 3 DOFs → describe the horizontal plane, longitudinal, and lateral models.
- 4 DOFs → describe the motion in the horizontal plane with heading autopilots and system damping
- 6 DOFs → describe the fully coupled equation of motion used for simulation and prediction.

Naval Architecture

- Maneuvering theory: moving at positive constant speed in calm water
- Seeking theory: motion of ships at zero or constraint speed in waves, which can be analyzed. The hydrodynamic coefficient and wave forces are computed as a function



Introduction

Tools and Toolboxes



Marine Systems Simulator (MSS)

Version 1.2 (74.5 MB) by Thor I. Fossen

Matlab library for the simulation of guidance, navigation and control systems for marine craft including ships, rigs, AUVs, ROVs and USVs
<https://github.com/cybergalactic/MSS>

[Follow](#)

[Share](#)

[Open in MATLAB Online](#)

★★★★★ (12)

4.3K Downloads

Updated 3 Dec 2024

[View License on GitHub](#)

Marine Systems Simulator (MSS) Toolbox¹

The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine systems. It includes hydrodynamic models for ships, underwater vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for real-time simulation. The algorithms and methods are described in:

T. I. Fossen (2021). Handbook of Marine Craft Hydrodynamics and Motion Control. 2nd. Edition, Wiley. ISBN-13: 978-1119575054

MATLAB Release Compatibility

Created with R2010a
Compatible with any release

Platform Compatibility

Windows macOS Linux

[mathworks-robotics /](#)

[modeling-and-simulation-of-an-AUV-in-Simulink](#)

Public

[Notifications](#)

[Fork 32](#)

[Star 73](#)

[Code](#)

[Issues 2](#)

[Pull requests](#)

[Actions](#)

[Projects](#)

[Security](#)

[Insights](#)

[master](#)

[?](#)

[Go to file](#)

[Code](#)

MATLAB Simscape Multibody²



cosorio-MLSL-applications

Update README.md

ae8638e · 3 years ago

[6 Commits](#)

[Source](#)

added Architecture folder

3 years ago

[resources/project](#)

added Architecture folder

3 years ago

About

This repository contains a variety of demonstration example models associated with the Design, Modeling and Simulation of Autonomous Underwater Vehicles webinar and video series.



¹T. I. Fossen and T. Perez (2004). Marine Systems Simulator (MSS). URL:
<https://github.com/cybergalactic/MSS>

²<https://github.com/mathworks-robotics/modeling-and-simulation-of-an-AUV-in-Simulink>

Overview of this seminar

- Fundamental knowledge
- 6-DOF model of a Rigid Body
- Ocean dynamics
- Environmental disturbances
- An example of the ODIN
- Guidance - Navigation - Control
- Numerical simulations
- Quasi-Physical Simulation for UV motion control
- An example for GNC-integrated Quasi-Physical simulation

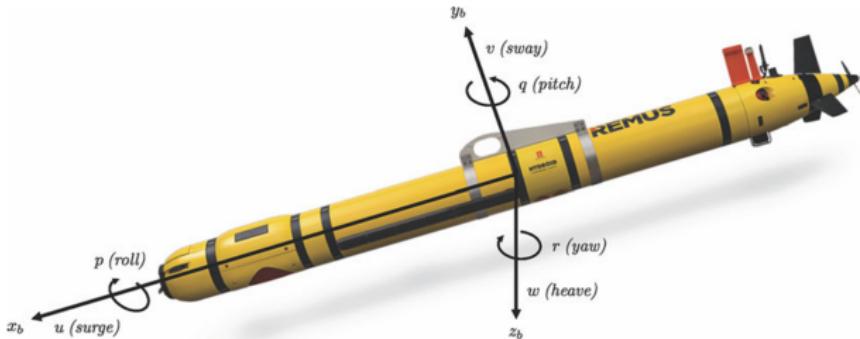


Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



Fundamental knowledge for modeling

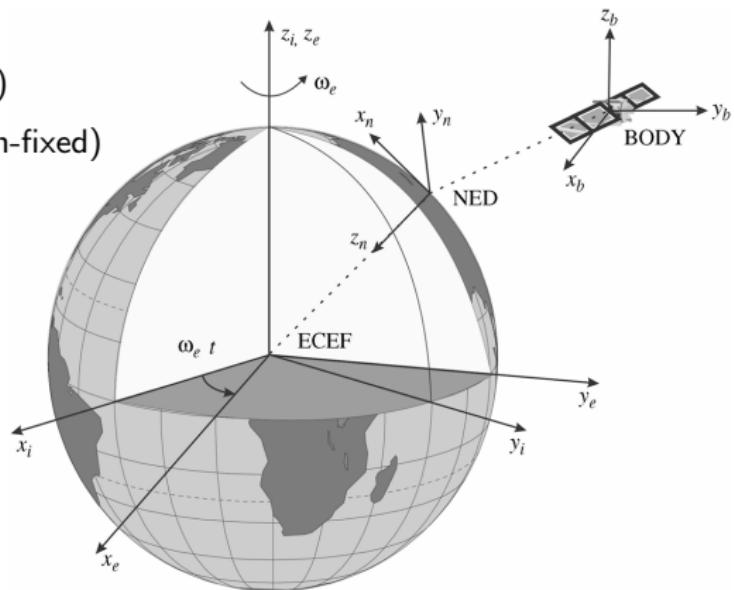
Coordinates

Earth-Centered reference frames

- ECI - $\{i\}$ - (Earth-centered inertial)
- ECEF - $\{e\}$ - (Earth-centered Earth-fixed)

Geographic reference frames

- NED - $\{n\}$ - (North-East-Down)
- BODY - $\{b\}$ - (Craft-fixed)

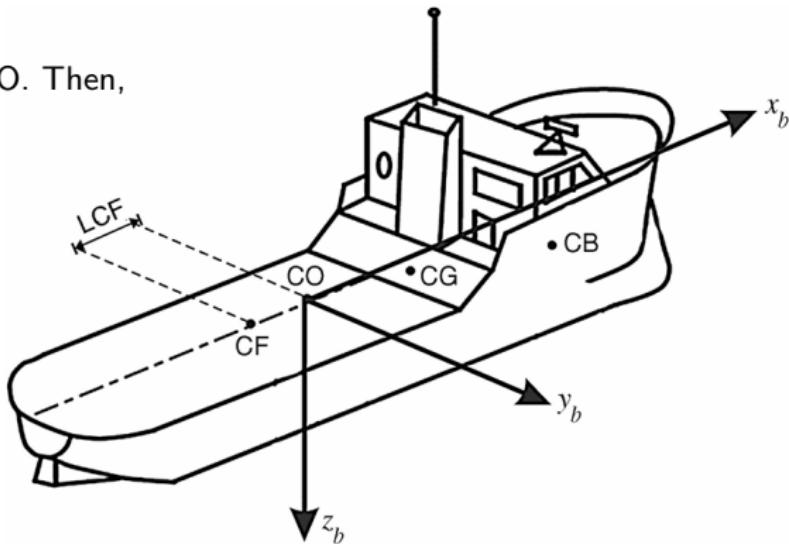


Fundamental knowledge for modeling

Coordinates

The origin of the BODY frame is CO. Then,

- CG - Center of Gravity
- CB - Center of Buoyancy
- CF - Center of flotation

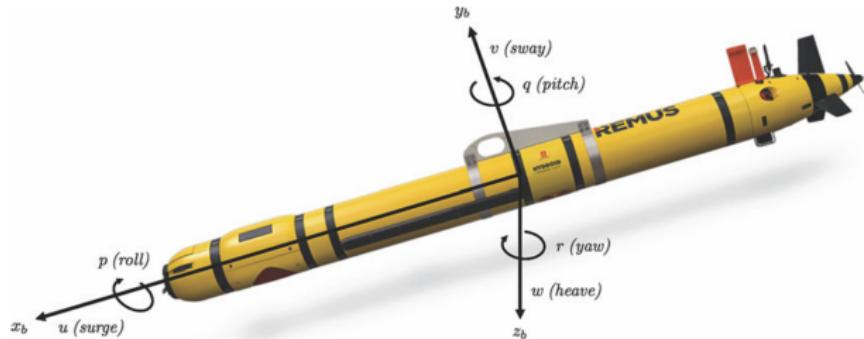


The center of flotation is the centroid of water plane area A_{wp} in calm water.



Fundamental knowledge for modeling

Notations



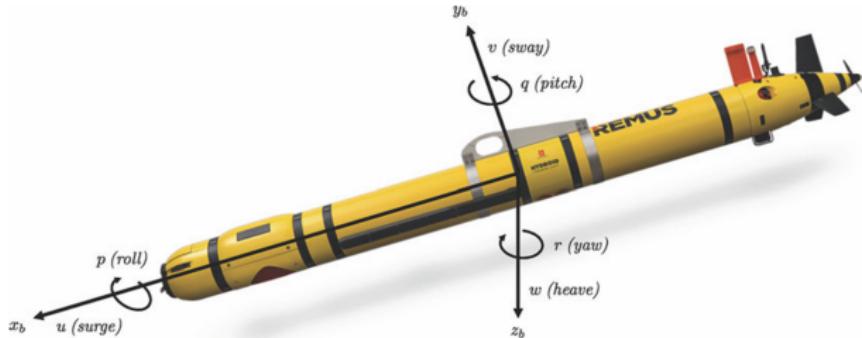
Notation of SNAME (1950)

- Motion of x -direction (surge) – x , controlled by X [N]
- Motion of y -direction (sway) – y , controlled by Y [N]
- Motion of z -direction (heave) – z , controlled by Z [N]
- Rotation of x -direction (roll) – ϕ , controlled by K [Nm]
- Rotation of y -direction (pitch) – θ , controlled by M [Nm]
- Rotation of z -direction (yaw) – ψ , controlled by N [Nm]



Fundamental knowledge for modeling

General coordinates



General coordinate

- $\eta = \begin{bmatrix} (p_{b/n}^n)^T & \Theta_{nb}^T \end{bmatrix}^T$, and $p_{b/n}^n = \begin{bmatrix} N & E & D \end{bmatrix}^T$, $\Theta_{nb} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$. In addition, the relative position of craft could be represented in frame ECI.
- $\nu = \begin{bmatrix} (v_{b/n}^b)^T & (\omega_{b/n}^b)^T \end{bmatrix}^T$, and $v_{b/n}^b = \begin{bmatrix} u & v & w \end{bmatrix}^T$, $\omega_{b/n}^b = \begin{bmatrix} p & q & r \end{bmatrix}^T$
- $\tau = \begin{bmatrix} \tau_1^T & \tau_2^T \end{bmatrix}^T$, and $\tau_1 = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$, $\tau_2 = \begin{bmatrix} K & M & N \end{bmatrix}^T$

Notice that η denotes the position and orientation vector with coordinates in the earth-fix frame. ν denotes the linear and angular velocity vector with coordinates in the body-fixed frame and τ denotes the force and torque acting on the vehicle.

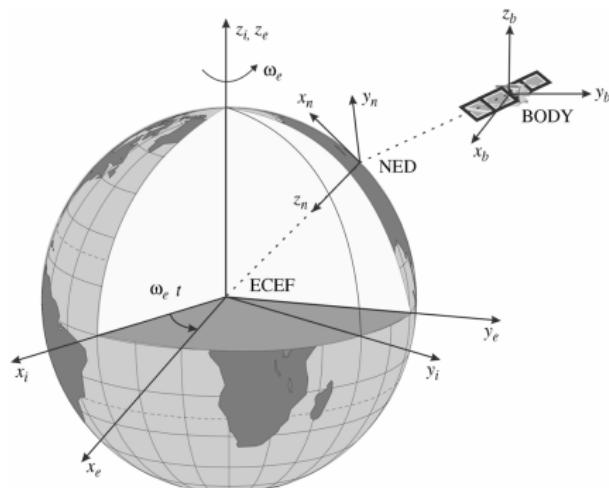


Fundamental knowledge for modeling

Euler angles

Frames transformation

- Euler angles
- Rotation matrices
- Skew-Symmetric of a matrix
- Cross product
- Euler's theorem on rotation
- Linear/Angular velocity transformation
- Rotation matrix differential
- ...



In another perspective, **quaternion** representation is also a good option!



Fundamental knowledge for modeling

Kinematics

The 6 DOFs and 3 DOFs kinematic equations can be expressed in the following

6 DOFs kinematic equations

$$\dot{\eta} = J_{\Theta}(\eta)\nu \quad (1)$$

that is equivalent to

$$\begin{bmatrix} \dot{p}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} = \begin{bmatrix} R_b^n & 0 \\ 0 & T_{\Theta}(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} v_{b/n}^b \\ \omega_{b/n}^b \end{bmatrix} \quad (2)$$

where R_b^n and $T_{\Theta}(\Theta_{nb})$ denote the rotation and transformation matrices.

3 DOFs kinematic equations

$$\dot{\eta} = R_z(\psi)\nu \quad (3)$$

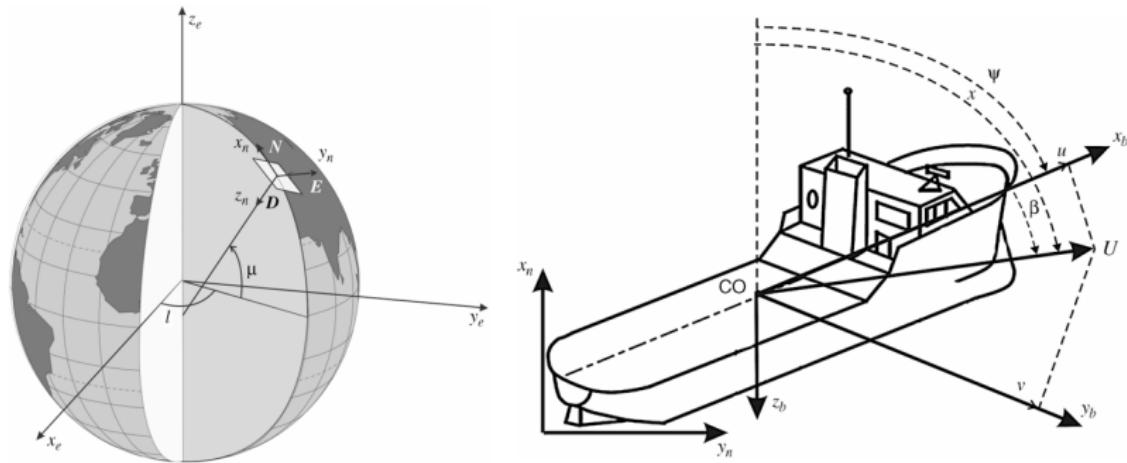
with $\eta = [x \ y \ \psi]^T$ and $\nu = [u \ v \ r]^T$

Fundamental knowledge for modeling

Other transformations

In addition, for more purposes, other transformations are needed

- Transformation between ECEF and NED: wide area or terrestrial guidance and navigation.
- Transformation between BODY and FLOW: express the hydrodynamic data

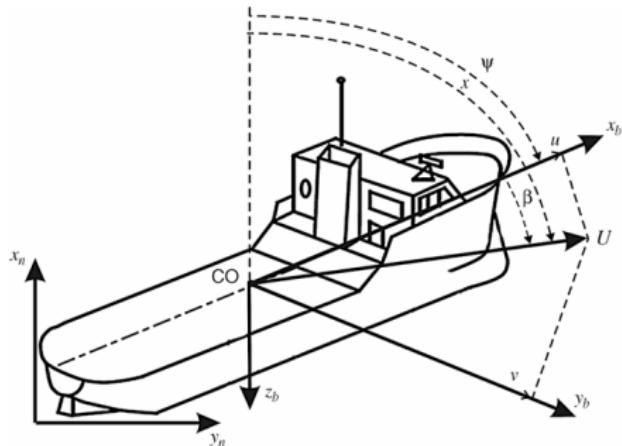


Fundamental knowledge for modeling

Transformation between BODY and FLOW

Similarly, in the first case, the craft is assumed to be moving on the horizon.

Definition of Course, Heading, and Sideslip Angles



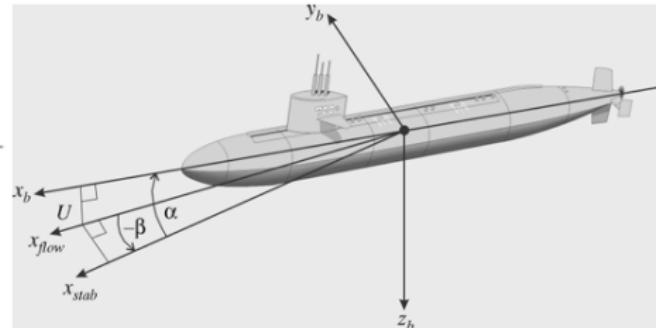
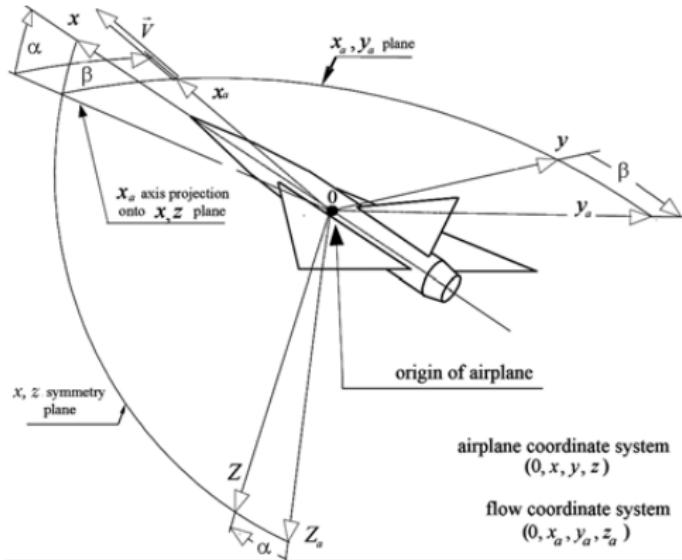
- Course angle χ : The rotation of x_n and **velocity vector** u of craft around z_n
- Heading angle ψ : Yaw angle of craft
- Sideslip (Drift) angle β : The rotation of x_b and **velocity vector** u of craft around z_b

Thus $\chi = \psi + \beta$.



Fundamental knowledge for modeling

Transformation between BODY and FLOW



BODY to FLOW transformation

$$R_b^{flow} = R_{z, -\beta} R_{y, \alpha} = \begin{bmatrix} \cos \beta \cos \alpha & \sin \beta & \cos \beta \sin \alpha \\ -\sin \beta \cos \alpha & \cos \beta & -\sin \beta \sin \alpha \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (4)$$

Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



6-DOF model of a Rigid Body

Goal

Final goal

To find the equations that describe the behavior of the system

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \quad (5)$$

in addition, as discussed above,

$$\dot{\eta} = J_{\Theta}(\eta)\nu \quad (6)$$

- Newton-Euler equations of motion about CG (Center gravity)
- Newton-Euler equations of motion about CO



6-DOF model of a Rigid Body

Fundamental

Newton's second law

The Newton-Euler formulation is based on Newton's second law, which relates mass m , acceleration $\dot{\vec{v}}_{g/i}$ and force \vec{f}_g

$$m\dot{\vec{v}}_{g/i} = \vec{f}_g \quad (7)$$

Euler's First and Second Axioms

The relationship between linear momentum and angular momentum

$$\frac{d}{dt}\vec{p}_g = \vec{f}_g \quad (8)$$

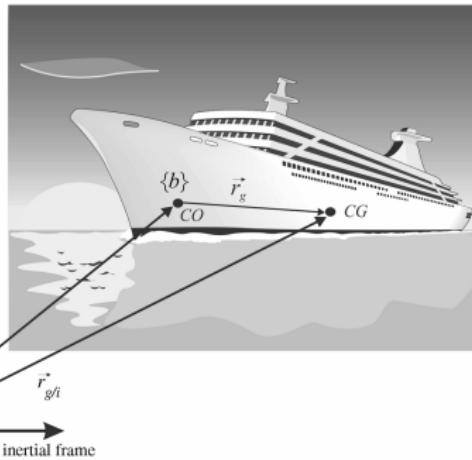
$$\frac{d}{dt}\vec{h}_g = \vec{m}_g \quad (9)$$

where \vec{p}_g and \vec{h}_g are the linear and angular momentum, respectively. \vec{f}_g and \vec{m}_g are the forces and moments acting on the CG.

6-DOF model of a Rigid Body

Fundamental

- Translational motion
- Rotational motion



Equations of Motion about CG

The Newton-Euler equation could be represented in the matrix form according to

$$M_{RB}^{CG} \begin{bmatrix} \dot{v}_{g/n}^b \\ \dot{\omega}_{b/n}^b \end{bmatrix} + C_{RB}^{CG} \begin{bmatrix} v_{g/n}^b \\ \omega_{b/n}^b \end{bmatrix} = \begin{bmatrix} f_g^b \\ m_g^b \end{bmatrix} \quad (10)$$



6-DOF model of a Rigid Body

Equations of Motion about CG

Translational motion

Using Euler's first axiom,

$$f_g = \frac{d}{dt}(mv_{g/i}) \quad (11)$$

by several tedious mathematical transformations

$$m\left(\dot{v}_{g/n}^b + \omega_{b/n}^b \times v_{g/n}^b\right) = f_g^b \quad (12)$$

Rotational motion

Using Euler's second axiom,

$$m_g = \frac{d}{dt}(I_g \omega_{b/i}) \quad (13)$$

by several tedious mathematical transformations

$$I\dot{\omega}_{b/n}^b - S(I_g \omega_{b/n}^b) \times \omega_{b/n}^b = m_g^b \quad (14)$$

6-DOF model of a Rigid Body

Equations of Motion about CO

By several SUPER TEDIOUS mathematical transformations

Nonlinear 6 DOF Rigid-Body Equations of Motion

$$m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X, \quad (15)$$

$$m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y, \quad (16)$$

$$m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(r\dot{p})] = Z, \quad (17)$$

$$I_x \dot{p} + (I_z - I_y)qr - (r + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K, \quad (18)$$

$$I_y \dot{q} + (I_x - I_z)rp - (p + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\ + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M, \quad (19)$$

$$I_z \dot{r} + (I_y - I_x)pq - (q + rp)I_{xz} + (q^2 - p^2)I_{xx} + (rq - \dot{p})I_{zx} \\ + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N. \quad (20)$$



6-DOF model of a Rigid Body

Matrix Form

Rigid-body dynamics model

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \quad (21)$$

where

$$M_{RB} = \begin{bmatrix} mI_{3 \times 3} & -mS(r_g^b) \\ mS(r_g^b) & I_b \end{bmatrix} \quad (22)$$

$$C_{RB} = \begin{bmatrix} 0_{3 \times 3} & -S(M_{11}\nu_1 + M_{12}\nu_2) \\ -S(M_{11}\nu_1 + M_{12}\nu_2) & -S(M_{21}\nu_1 + M_{22}\nu_2) \end{bmatrix} \quad (23)$$



6-DOF model of a Rigid Body

Linearized 6 DOF Rigid body Equation of Motion



Dynamics model

Simplified 6 DOF Rigid body Equation of Motion

- The Origin CO coincides with the CG
- Translation of the origin CO such that I_b becomes diagonal

Simplified 6 DOF Rigid body Equation of Motion

$$m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X, \quad (24)$$

$$m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y, \quad (25)$$

$$m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = Z, \quad (26)$$

$$I_x\dot{p} + (I_z - I_y)qr + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K, \quad (27)$$

$$I_y\dot{q} + (I_x - I_z)rp + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M, \quad (28)$$

$$I_z\dot{r} + (I_y - I_x)pq + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N. \quad (29)$$



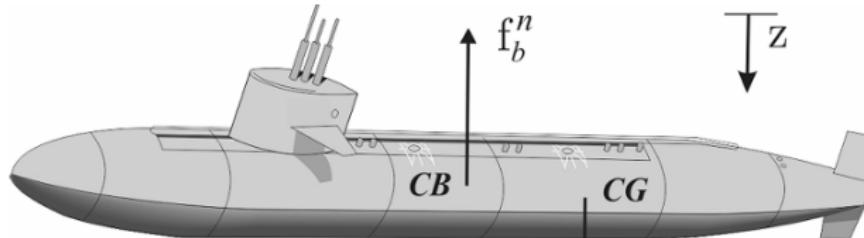
Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



Ocean dynamics

Hydrostatics



Forces act in the vertical plane

$$f_g^n = \begin{bmatrix} 0 & 0 & W \end{bmatrix}^T; \quad f_b^n = -\begin{bmatrix} 0 & 0 & B \end{bmatrix}^T \quad (30)$$

Thus, the gravity vector is

$$g(\eta) = - \begin{bmatrix} f_g^b + f_b^b \\ r_g^b \times f_g^b + r_b^b \times f_b^b \end{bmatrix} \quad (31)$$

$$= - \begin{bmatrix} R_b^n(\Theta_{nb})^{-1} (f_g^n + f_b^n) \\ r_g^b \times R_b^n(\Theta_{nb})^{-1} f_g^n + r_b^b \times R_b^n(\Theta_{nb})^{-1} f_b^n \end{bmatrix}. \quad (32)$$

Note: These formulations become more complex when considering the surface vessels



Ocean dynamics

Seakeeping Theory(wave and wind)

Vehicle-Ocean dynamics model

$$\dot{\eta} = J_{\Theta}(\eta)\nu$$

$$M_{RB}\dot{\nu} + C_{RB}^*\nu + M_A\dot{\nu}_r + C_A^*\nu_r + (D_P + D_V)\nu_r + \mu + G\eta + g_o = \tau_{\text{wind}} + \tau_{\text{wave}} + \tau$$

Vehicle-Ocean dynamics model

Inertia forces: $M_{RB}\dot{\nu} + C_{RB}^*\nu + M_A\dot{\nu}_r + C_A^*\nu_r$

Damping forces: $+ (D_P + D_V)\nu_r + \mu$

Restoring forces: $+ G\eta + g_o$

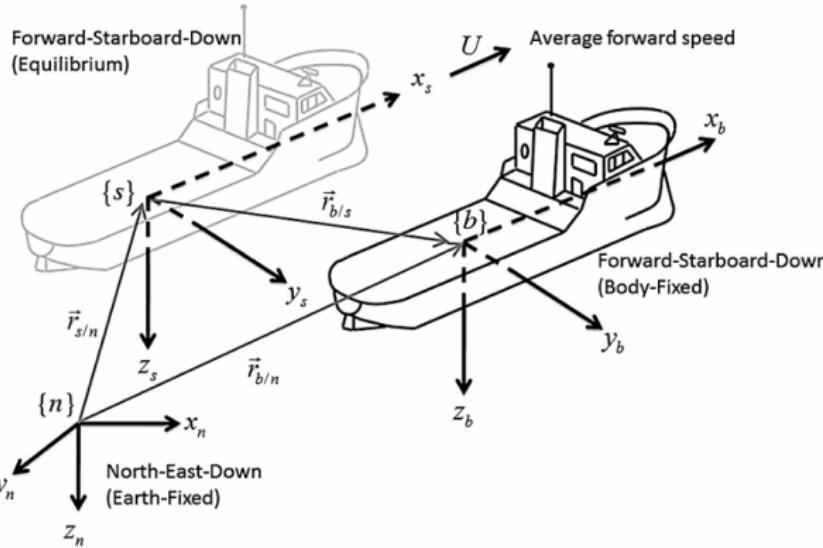
Wind and wave forces: $= \tau_{\text{wind}} + \tau_{\text{wave}}$

Propulsion forces: $+ \tau$



Ocean dynamics

Seakeeping kinematics (wave and wind)



Several notations

$$\bullet v_{s/n}^n = \begin{bmatrix} U \cos \bar{\psi} & U \sin \bar{\psi} & 0 \end{bmatrix}^\top$$

$$\bullet \omega_{s/n}^n = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$$

$$\bullet \Theta_{ns} = \begin{bmatrix} 0 & 0 & \bar{\psi} \end{bmatrix}^\top$$



Ocean dynamics

Seakeeping kinematics(wave and wind)

Define the perturbation coordinate

$$\delta\eta = \begin{bmatrix} r_{b/s}^s \\ \Theta_{sb} \end{bmatrix}; \quad \delta\nu = \begin{bmatrix} v_{b/s}^b \\ \omega_{b/s}^b \end{bmatrix}; \quad \xi = \delta\eta; \quad \Theta_{sb} = \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \quad (33)$$

Thus, the following equations are achieved as the transformation between BODY and SEAKEEPING coordinates

Transformation between BODY and SEAKEEPING

$$\dot{\nu} = -UL\delta\nu + \delta\dot{\nu} \quad (34)$$

where U is the velocity of the vehicle and $L \in \mathbb{R}_{6 \times 6}$

Linear transformations

$$\delta\nu \approx \nu + U(L\delta\eta - e_1) \quad (35)$$

$$\delta\dot{\nu} \approx \dot{\nu} + UL\nu \quad (36)$$

Ocean dynamics

Added mass(wave and wind)

From Cummins Equation in SEAKEEPING Coordinates, the following equations are introduced

Frequency-domain seakeeping equation of motion

$$\left(-\omega^2(M_{RB} + \bar{A}(\omega)) - j\omega B(\omega) + C \right) \ddot{\xi}(j\omega) = \tau_{wind}(j\omega) + \tau_{wave}(j\omega) + \delta\tau(j\omega) \quad (37)$$

Time-domain seakeeping equation of motion

$$(M_{RB} + \bar{A}) \ddot{\xi} + \int_{-\infty}^t \bar{K}(t - \tau) \dot{\xi}(t) d\tau + \bar{C}\xi = \tau_{wind} + \tau_{wave} + \delta\tau \quad (38)$$

$$\text{where } \bar{K} = \frac{2}{\pi} \int_0^\infty B(\omega) \cos(\omega t) d\omega$$

Notice: For underwater vehicle, $A(\omega) = \text{const}$ and $B(\omega) = 0$



Ocean dynamics

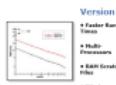
Added mass(wave and wind)

Hydrodynamic added mass forces and moments in 6 DOFs

- The expressions are complicated and not suited for control design.
- Hydrodynamic software programs such as WAMIT and ShipX can be used to compute the added mass term.



[Home](#)
[Technical Description](#)
[Version 7](#)
[Licensing Information](#)
[Downloads](#)
[User Manual 7](#)
[User Manual 6.4B](#)
[Demonstrations Program](#)
[Examples of Structures](#)
[Publications](#)
[Company Info](#)
[Contact Us](#)
[Consultants](#)
[News](#)
[Links](#)



WAMIT® is the most advanced set of tools available for analyzing wave interactions with offshore platforms and other structures or vessels. Since the announcement of Version 1.0 in 1987 at PTC, WAMIT has gained widespread recognition for its ability to analyze complex structures with a high degree of accuracy and efficiency. WAMIT® is used worldwide. WAMIT has been adopted by more than 100 industrial and research organizations worldwide. Upgrade license fees start at approximately one-year intervals with major enhancements.

WAMIT version 6, which was available from 2000-2011, implemented a quantum leap in capabilities, including a faster method of analysis and various options for defining the geometry. Version 6 has the additional capability of complete second-order nonlinear analysis in bidirectional and bidirectional wave environments. Version 7 adds the ability to analyze complex structures with a high degree of accuracy and efficiency. WAMIT® is used worldwide. WAMIT has been adopted by more than 100 industrial and research organizations worldwide. Upgrade license fees start at approximately one-year intervals with major enhancements.

WAMIT version 7 has additional extensions and updates from version 6.6. The computations can be performed using paralel processing on PC's with multiple processors (MPUs), and the increased memory size of contemporary PCs can be exploited to achieve substantially faster run times for large complex applications. A detailed description of the changes in the v7 User Manual. Most of these options can be tested using the Demonstrations Version of WAMIT®.

WAMIT version 7 is available in two separate versions for Windows and Linux operating systems. Licensing information

Examples of structures analyzed by WAMIT

(for more details click on one of the structures)



[Our services](#) > [Software](#) >

SOFTWARE



For more than 15 years we have developed a hydrodynamic workbench called ShipX. ShipX incorporates a multitude of hydrodynamic tools enabling the user to perform many different types of calculations based on the same vessel input.

Contact person



Edvard Ringen
Senior Software Developer
+47 92 80 30 06
Edvard.Ringen@sintef.no



From Cummins Equation in SEAKEEPING Coordinates, the following equations are introduced

Linear equation of motion (Zero-Speed Potential Coefficients)

$$M\dot{\nu} + C_{RB}^*\nu + C_A^*\nu_r + D\nu_r + \int_0^t K(t-\tau)[\nu(\tau) - Ue_1]d\tau + G\eta = \sum \tau \quad (39)$$

Linear equation of motion (Speed-Dependent Potential Coefficients)

$$M_U\dot{\nu} + C_{RB}^*\nu + C_A^*\nu_r + D_U\nu_r + \int_0^t K_U(t-\tau)[\nu(\tau) - Ue_1]d\tau + G\eta = \sum \tau \quad (40)$$

where $\sum \tau = \tau_{wind} + \tau_{wind} + \tau$



Goal

$$\underbrace{M_{RB}\dot{\nu} + C_{RB}(\nu)\nu}_{\text{rigid-body forces}} + \underbrace{M_A\dot{\nu}_r + C_A(\nu_r)\nu_r + D(\nu_r)\nu_r}_{\text{hydrodynamic forces}} + \underbrace{g(\eta) + g_o}_{\text{hydrostatic forces}} = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \quad (41)$$

or

$$M\dot{\nu}_r + C(\nu_r)\nu_r + D(\nu_r)\nu_r + g(\eta) + g_o = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \quad (42)$$

where $\nu_r = \nu - \nu_c$ is the relative velocity vector.

- For slender bodies, such as AUV, the longitudinal (surge, heave, and pitch motion) and lateral (sway, roll, and yaw motion) models are referred to consider.
- For ROVs operating, the horizontal-plane (surge, sway, and yaw motion) and depth (heavy) models are referred to consider.



3DOF maneuvering model

The horizontal motions (surge, sway, and heave) of a marine craft moving at forward speed could be described by a zero frequency model, where

$$M_A = \begin{bmatrix} A_{11}(0) & 0 & 0 \\ 0 & A_{22}(0) & A_{26}(0) \\ 0 & A_{62}(0) & A_{66}(0) \end{bmatrix} \quad (43)$$

$$D_p = 0 \quad (44)$$

are the constant matrices.

Limitation One limitation of the zero-frequency assumption is that it cannot be applied to heave, roll, and pitch. For 2nd-order mass-damper-spring systems the dominating frequencies are the natural frequencies.



Extension to 6-DOF models

Solution: Formulate frequency-independent models in heave, pitch, and roll at their respective natural frequencies and not the zero frequency.

$$\omega_3 = \sqrt{\frac{C_{33}}{m + A_{33}(\omega_3)}} \quad (45)$$

$$\omega_4 = \sqrt{\frac{C_{44}}{I_x + A_{44}(\omega_4)}} \quad (46)$$

$$\omega_5 = \sqrt{\frac{C_{55}}{I_y + A_{55}(\omega_5)}} \quad (47)$$

Key assumption: No coupling between the surge-sway-yaw and the heave-roll-pitch subsystems.



Ocean dynamics

Added Mass Forces in a Rotating Coordinate System

The expression for the fluid kinetic energy T_A , see Ref.³, can be written as a quadratic form of the body axis velocity vector components, that is:

$$T_A = \frac{1}{2} \boldsymbol{\nu}^T M_A \boldsymbol{\nu} \quad (48)$$

Here M_A is a 6×6 added inertia matrix defined as:

$$M_A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \triangleq - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}, \quad Y_A = Y_{\dot{u}} \dot{u}, \quad Y_{\dot{u}} \triangleq \frac{\partial Y}{\partial \dot{u}} \quad (49)$$

Notice: In a real fluid, these 36 constants could be distinct. With ideal (without friction), $M_A = M_A^\top$

³Imlay, Frederick H. *The complete expressions for added mass of a rigid body moving in an ideal fluid.* No. Report 1528. 1961.



Ocean dynamics

Added Mass Forces in a Rotating Coordinate System

Hydrodynamic Coriolis-Centripetal Matrix $C_A(\nu)$

For a rigid body moving through an ideal fluid the hydrodynamic Coriolis and centripetal matrix $C_A(\nu)$ could always be parametrized such that it is skew-symmetric

$$C_A(\nu) = -C_A(\nu)^\top, \quad \forall \nu \quad (50)$$

As mentioned above,

$$C_A(\nu) = \begin{bmatrix} 0 & -S(A_{11}\nu_1 + A_{12}\nu_2) \\ -S(A_{11}\nu_1 + A_{12}\nu_2) & -S(A_{21}\nu_1 + A_{22}\nu_2) \end{bmatrix} \quad (51)$$



Ocean dynamics

Added Mass Forces in a Rotating Coordinate System

Vehicle-Ocean dynamics model

$$\dot{\eta} = J_{\Theta}(\eta)\nu$$

$$M_{RB}\dot{\nu} + C_{RB}^*\nu + M_A\dot{\nu}_r + C_A^*\nu_r + (D_P + D_V)\nu_r + \mu + G\eta + g_o = \tau_{wind} + \tau_{wave} + \tau$$

Vehicle-Ocean dynamics model

Inertia forces: $M_{RB}\dot{\nu} + C_{RB}^*\nu + M_A\dot{\nu}_r + C_A^*\nu_r$

Damping forces: $+ (D_P + D_V)\nu_r + \mu$

Restoring forces: $+G\eta + g_o$

$$\text{Wind and wave forces:} \quad = \tau_{\text{wind}} + \tau_{\text{wave}}$$

Propulsion forces: $+ \tau$



In practical implementations, it is difficult to determine higher-order terms as well as the off-diagonal terms in general expression for hydrodynamic damping.

The different damping terms contribute to both linear and quadratic damping. However, it is in general difficult to separate these effects. In many cases, it is convenient to write total hydrodynamic damping as

$$D(\nu_r) = D + D_n(\nu_r) \quad (52)$$

- Linear viscous damping

$$D = D_p + D_v \approx -\text{diag}(X_u, Y_v, Z_w, K_p, M_q, N_r) \quad (53)$$

- Nonlinear surge damping

- Nonlinear Surge Damping: Damping Due to Vortex Shedding
- Cross-Flow Drag Principle (lifting Forces): For relative current angles, the cross-flow drag principle may be applied to calculate the nonlinear damping force in sway and the yaw moment



The approximate diagonal matrix D_p is expressed as:

$$D_p \approx \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{33}(\omega_{\text{heave}}) & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{44}(\omega_{\text{roll}}) & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{55}(\omega_{\text{pitch}}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (54)$$

The natural frequencies ω_{heave} , ω_{roll} , and ω_{pitch} can be computed as introduced in the above.

A diagonal matrix usually approximates the linear viscous damping terms:

$$D_v \approx \text{diag}\{B_{11v}, B_{22v}, B_{33v}, B_{44v}, B_{55v}, B_{66v}\} \quad (55)$$

where the elements B_{iiv} ($i = 1, \dots, 6$) can be computed from the time constants and natural periods of the system (see Section 6.4).



Ocean dynamics

Linear viscous damping

The expressions for the diagonal elements of D_v are as follows:

$$B_{11v} = \frac{m + A_{11}(0)}{T_{\text{surge}}} = \frac{8\pi\zeta_{\text{surge}}[m + A_{11}(0)]}{T_{n,\text{surge}}} \quad (56)$$

$$B_{22v} = \frac{m + A_{22}(0)}{T_{\text{sway}}} = \frac{8\pi\zeta_{\text{sway}}[m + A_{22}(0)]}{T_{n,\text{sway}}} \quad (57)$$

$$B_{33v} = 2\Delta\zeta_{\text{heave}}\omega_{\text{heave}}[m + A_{33}(\omega_{\text{heave}})] \quad (58)$$

$$B_{44v} = 2\Delta\zeta_{\text{roll}}\omega_{\text{roll}}[I_x + A_{44}(\omega_{\text{roll}})] \quad (59)$$

$$B_{55v} = 2\Delta\zeta_{\text{pitch}}\omega_{\text{pitch}}[I_y + A_{55}(\omega_{\text{pitch}})] \quad (60)$$

$$B_{66v} = \frac{I_z + A_{66}(0)}{T_{\text{yaw}}} = \frac{8\pi\zeta_{\text{yaw}}[I_z + A_{66}(0)]}{T_{n,\text{yaw}}} \quad (61)$$



Quadratic drag and lifting in 6DOF

Quadratic drag and lifting in 6DOF

$$D_n(\nu_r)\nu_r = [|\nu_r|^\top D_i(\rho, A, C_D)\nu_r]_{6 \times 1} \quad (62)$$

The viscous damping force due to vortex shedding can be modeled as

$$f(U) = -\frac{1}{2}\rho C_D A |U_r| U_r \quad (63)$$

where ρ is the fluid density, C_D is drag coefficient, A is the projected cross-sectional area, and U is the velocity of the vehicle.



Hydrodynamic Mass–Damper

- **Added mass** M_A due to the inertia of the surrounding fluid (see Section 6.2). The corresponding Coriolis and centripetal matrix due to added mass is due to the rotation of $\{b\}$ with respect to $\{n\}$ and is denoted $C_A(\nu_r)$ (see Section 6.3).
- **Radiation-induced potential damping** D_p due to the energy carried away by generated surface waves.
- **Viscous damping** caused by skin friction, wave drift damping, vortex shedding, and lift/drag (see Section 6.4).

The resulting hydrodynamic force is written as:

$$\tau_{\text{hyd}} = -M_A \dot{\nu}_r - C_A(\nu_r) \nu_r - D_p \nu_r + \tau_{\text{visc}} \quad (64)$$

where $\nu_r = \nu - \nu_c$ with $\nu_c = [u_c, v_c, w_c, 0, 0, 0]^\top$ is the relative velocity due to an *irrotational constant ocean current* (see Section 8.3), and

$$\tau_{\text{visc}} = -D_v \nu_r - D_n(\nu_r) \nu_r \quad (65)$$

Hydrostatic Spring Stiffness: Restoring forces due to Archimedes

$$\tau_{\text{hs}} = -g(\eta) - g_o$$



Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



The environmental disturbances include

- Wind forces and moments
- Wave forces and moments

In this context, if the autonomous underwater vehicles are considered to be working at the deep water level, the wind and wave disturbances are not significant enough to be considered.

The wind and wave disturbances become more significant when we consider a surface vehicle, and it could be introduced in the future version.

[Coming soon!!]



Table of Contents

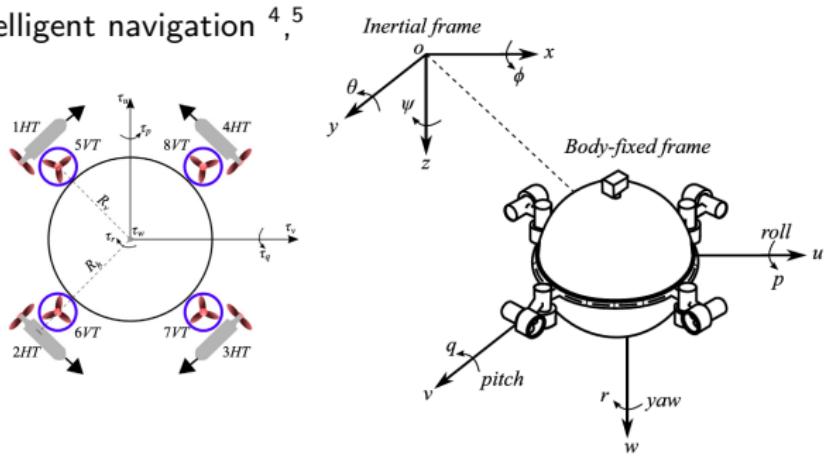
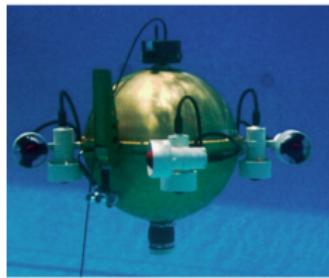
- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



An example of the ODIN

Rigid body equation of motion

ODIN = Omni-directional intelligent navigation^{4, 5}



⁴Choi, Hyun-Taek, et al. "Development of an underwater robot, ODIN-III." Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). Vol. 1. IEEE, 2003.

⁵Ali, Nihad, Isaac Tawiah, and Weidong Zhang. "Finite-time extended state observer based nonsingular fast terminal sliding mode control of autonomous underwater vehicles." Ocean Engineering 218 (2020): 108179. 

An example of the ODIN

Rigid body equation of motion

Dynamic equations

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \quad (67)$$

where

$$M_{RB} = \begin{bmatrix} mI_{3 \times 3} & -mS(r_g^b) \\ mS(r_g^b) & I_b \end{bmatrix}; C_{RB} = \begin{bmatrix} mS(\nu_2) & -mS(\nu_2)S(r_g^b) \\ mS(\nu_2)S(r_g^b) & -S(I_b\nu_2) \end{bmatrix}$$

Notice that, the term $C_{RB}(\nu)$ is modified by several mathematical transformations such as $C_{RB}(\nu) = C_{RB}(\nu_r)$ by linear velocity-independent parametrizations. Thus, the above dynamic equation could be represented as

$$M_{RB}\dot{\nu}_r + C_{RB}\nu_r = \tau_{RB} \quad (68)$$

with ν_r is the relative velocity of the vehicle. Therefore, the kinematics equation is represented as

$$\dot{\eta} = J(\eta)\nu$$

where $J(\eta)$ is the Jacobian matrix.



An example of the ODIN

Added mass

In order to lose the general, let's consider the ODIN as a spherical solid. Thus, the ODIN-geometry could be described as the following equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 0 \quad (70)$$

Let's define ⁶

$$\alpha_0 = abc \int_0^\infty \frac{d\lambda}{(a^2 + \lambda)\Delta}, \quad \beta_0 = abc \int_0^\infty \frac{d\lambda}{(b^2 + \lambda)\Delta}, \quad \gamma_0 = abc \int_0^\infty \frac{d\lambda}{(c^2 + \lambda)\Delta} \quad (71)$$

$$\text{with } \Delta = \{(a^2 + \lambda)(b^2 + \lambda)(c^2 + \lambda)\}^{\frac{1}{2}}$$

Special case: $a = b = c = R$, it is easy to achieve that

$$\alpha_0 = \beta_0 = \gamma_0 = R^3 \int_0^\infty \frac{d\lambda}{(R^2 + \lambda)^2 \sqrt{(R^2 + \lambda)}} = R^3 \frac{2}{3R^3} = \frac{2}{3} \quad (72)$$



⁶Lamb, Horace. Hydrodynamics. University Press, 1924.

An example of the ODIN

Added mass

Following ⁷, the added mass of the spherical-solid underwater vehicle could be represented as

$$X_{\dot{u}} = -\frac{\alpha_0}{2 - \alpha_0} \frac{4}{3} \pi \rho abc, \quad (73)$$

$$Y_{\dot{v}} = -\frac{\beta_0}{2 - \beta_0} \frac{4}{3} \pi \rho abc, \quad (74)$$

$$Z_{\dot{w}} = -\frac{\gamma_0}{2 - \gamma_0} \frac{4}{3} \pi \rho abc, \quad (75)$$

$$K_{\dot{p}} = -\frac{1}{5} \frac{(b^2 - c^2)(\gamma_0 - \beta_0)}{2(b^2 - c^2) + (b^2 + c^2)(\beta_0 - \gamma_0)} \frac{4}{3} \pi \rho abc, \quad (76)$$

$$M_{\dot{q}} = -\frac{1}{5} \frac{(c^2 - a^2)(\alpha_0 - \gamma_0)}{2(c^2 - a^2) + (c^2 + a^2)(\gamma_0 - \alpha_0)} \frac{4}{3} \pi \rho abc, \quad (77)$$

$$N_{\dot{r}} = -\frac{1}{5} \frac{(a^2 - b^2)(\beta_0 - \alpha_0)}{2(a^2 - b^2) + (a^2 + b^2)(\alpha_0 - \beta_0)} \frac{4}{3} \pi \rho abc. \quad (78)$$



⁷Imlay, Frederick H. The complete expressions for added mass of a rigid body moving in an ideal fluid. No. Report 1528. 1961.

An example of the ODIN

Added mass

Special case: $a = b = c = R$, then $\alpha_0 = \beta_0 = \gamma_0 = \frac{2}{3}$, therefore

$$X_{\dot{u}} = Y_{\dot{v}} = Z_{\dot{w}} = -\rho \frac{2}{3} \pi R^3 \quad (79)$$

$$K_{\dot{p}} = K_{\dot{p}} = K_{\dot{p}} = 0 \quad (80)$$

Written in matrix form,

$$M_A = \begin{bmatrix} \rho \frac{2}{3} \pi R^3 I_3 & 0_3 \\ 0_3 & 0_3 \end{bmatrix} \quad (81)$$

Then, it is easy to find the Coriolis term of added mass, which is not depended on ν_1

$$C_A(\nu_r) = \rho \frac{2}{3} \pi R^3 \begin{bmatrix} 0 & 0 & 0 & 0 & w_r & -v_r \\ 0 & 0 & 0 & -w_r & 0 & u_r \\ 0 & 0 & 0 & v_r & -u_r & 0 \\ 0 & w_r & -v_r & 0 & 0 & 0 \\ -w_r & 0 & u_r & 0 & 0 & 0 \\ v_r & -u_r & 0 & 0 & 0 & 0 \end{bmatrix} \quad (82)$$



An example of the ODIN

Gravitational/buoyancy forces

Hydrostatics force of Submerged Vehicles is represented as

$$g(\eta) = \begin{bmatrix} (W - B) \sin(\theta) \\ -(W - B) \cos(\theta) \sin(\phi) \\ -(W - B) \cos(\theta) \cos(\phi) \\ -(y_g W - y_b B) \cos(\theta) \cos(\phi) + (z_g W - z_b B) \cos(\theta) \sin(\phi) \\ (z_g W - z_b B) \sin(\theta) + (x_g W - x_b B) \cos(\theta) \cos(\phi) \\ -(x_g W - x_b B) \cos(\theta) \sin(\phi) - (y_g W - y_b B) \sin(\theta) \end{bmatrix} \quad (83)$$

with (x_g, y_g, z_g) and (x_b, y_b, z_b) are the coordinates of CO and CB, respectively.

Special case:

- $W = B \rightarrow$ neutral buoyant, the Gravitational/buoyancy forces just cause the rotations (moments).
- Oz_b and Oz_g are coincidence. That mean $x_g = x_b$ and $y_g = y_b$.
- $CB \equiv CG \equiv CO$ and $W = B \rightarrow g(\eta) = (0, 0, 0, 0, 0, 0)^\top$.



An example of the ODIN

Damping

Damping includes

- Linear damping

$$D = -\text{diag}(X_u, Y_v, Z_w, K_p, M_q, N_r) \quad (84)$$

- Nonlinear term (important for high-speed vehicles)

$$d(\nu_r) = \begin{bmatrix} \frac{1}{2}V_r^2 SC_D(\alpha) \cos \alpha - \frac{1}{2}V_r^2 SC_L(\alpha) \sin \alpha \\ Y_{cross} \\ -\frac{1}{2}V_r^2 SC_D(\alpha) \sin \alpha - \frac{1}{2}V_r^2 SC_L(\alpha) \cos \alpha \\ 0_{3 \times 1} \end{bmatrix} \quad (85)$$

More details, C_D , C_L are the drag, lifting coefficient, respectively.

$$F_{drag} = -\frac{1}{2}\rho V_r^2 SC_D(\alpha) \quad (86)$$

$$F_{lift} = -\frac{1}{2}\rho V_r^2 SC_L(\alpha) \quad (87)$$

$$Y_{cross} = -\frac{1}{2}\rho \int_{-L/2}^{L/2} T(x) C_d^{2D}(x) |v_r + xr| (v_r + xr) dx$$



Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



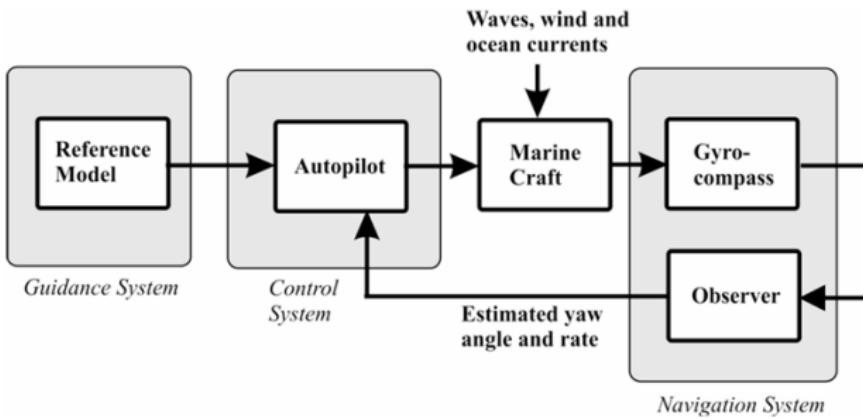
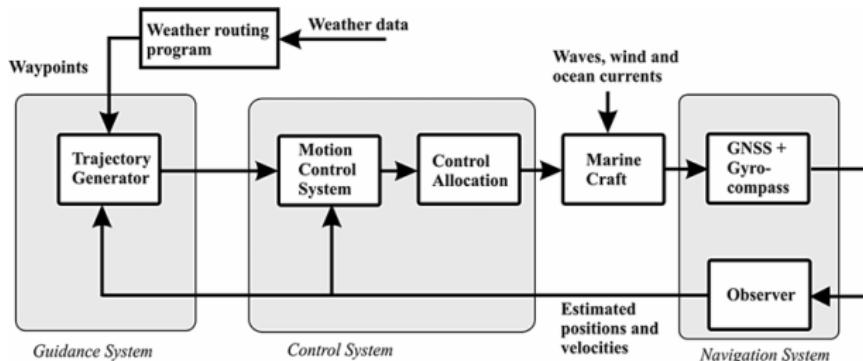
In the previous section, we introduce the modeling and mathematical model to describe the dynamics behavior. In the following steps, the control techniques, which include guidance, navigation and motion control, for a specific underwater vehicle (in general, consider the ocean vehicle). The theory and cases studies are organized as three independent part:

- **G - Guidance Systems:** Systems for automatically guiding the path of a marine craft, usually without direct or continuous human control.
- **N - Navigation Systems:** Systems for determination of the craft's position/attitude, velocity and acceleration.
- **C - Control Systems:** PID design methods for automatic control of position/attitude, velocity and acceleration. This involves control systems for stabilization, trajectory-tracking and path following control of marine craft.



Guidance - Navigation - Control

GNC overview



Guidance - Navigation - Control

Setpoint regulation, trajectory-tracking control and path-following control

Setpoint Regulation: The most basic guidance system is a constant input or setpoint provided by a human operator. The corresponding controller will then be a regulator.

Examples of setpoint regulation are constant depth, trim, heel and speed control. It could also be regulation to zero, which is commonly required in roll and pitch for instance.

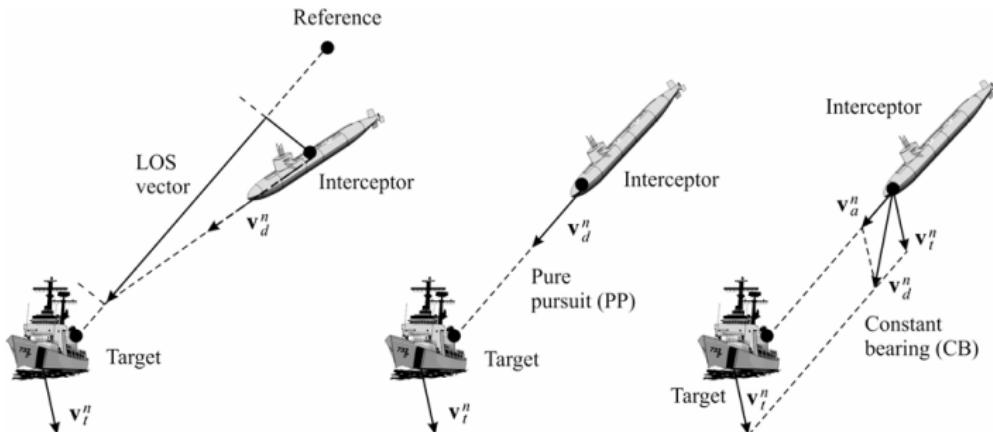
Trajectory-Tracking Control: The position and velocity of the marine craft should track desired time varying position and velocity reference signals. The corresponding feedback controller is a trajectory tracking controller. Tracking control can be used for course-changing maneuvers, speed-changing and attitude control. An advanced guidance system computes optimal time-varying trajectories from a dynamic model for a predefined control objective. If a constant setpoint is used as input to a low-pass filter (reference model) in an open-loop guidance system, the outputs of the filter will be smooth time-varying reference trajectories for position, velocity and acceleration (PVA).

Path-Following Control: This is to follow a predefined path independent of time (no temporal constraints). Moreover, no restrictions are placed on the temporal propagation along the path. This is typical for ships in transit between continents or underwater vehicles used to map the seabed.



Guidance - Navigation - Control

Guidance - Target tracking



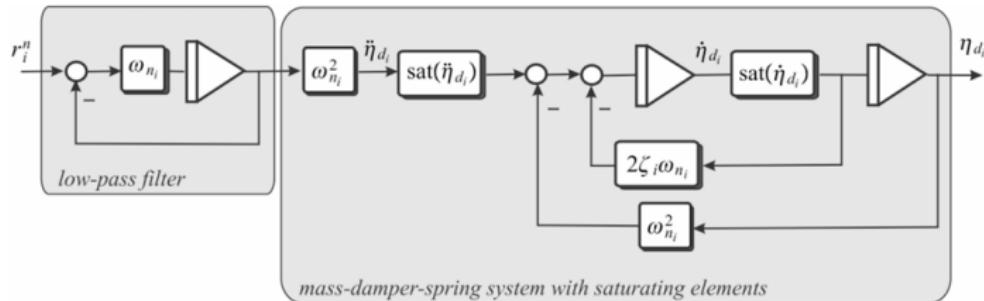
Several guidance strategies

- Line-of-Sight Guidance (LOS)
- Pure Pursuit Guidance (PP)
- Constant Bearing Guidance (CB)
- ...



Guidance - Navigation - Control

Guidance - Trajectory tracking



The easiest approach is that the desired trajectory can be computed using reference models generated by low-pass filters

$$\frac{x_d}{r}(s) = h_{lp}(s) = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \quad (89)$$

where $r(s)$ is set up by operator and $x_d(s)$ is the controller input.



The second one is that we consider the simulation model to generate the trajectory. For instance, a dynamic model could be chosen as

$$\dot{\eta}_d = J(\eta)\nu_d \quad (90)$$

$$M\dot{\nu}_d + N\nu_d + g(\nu_d) = \tau \quad (91)$$

The smooth trajectory could be generated by simulating the above dynamic model with a certain controller, such as PD control. To explain,

$$\tau = g(\eta_d) - J^{-1}(\eta_d)(K_p(\eta_d - \eta_{ref}) + Kd\dot{\eta}_d) \quad (92)$$



The vehicles' trajectory could be generated by optimization problems. The easiest approach method for this problem by consider a cost function of power, or time, and the purpose is to minimize this cost function. That mean,

$$x_d^* = \operatorname{argmin}_{x_d} (f(x, t)) \quad (93)$$

$$\text{s.t. } x_d \in \mathcal{X} \quad (94)$$

For examples, by considering the discrete model of vehicles, the optimization problem could be expressed in detail as follows

$$x_d^* = \operatorname{argmin}_{x_d} \sum x_d(k)^\top Q x_d(k) \quad (95)$$

or (96)

$$x_d^* = \operatorname{argmin}_{x_d} t \quad (97)$$

with constraint $x_d(k+1) = f(x_d(k), u(k))$ and $Q = Q^\top > 0$

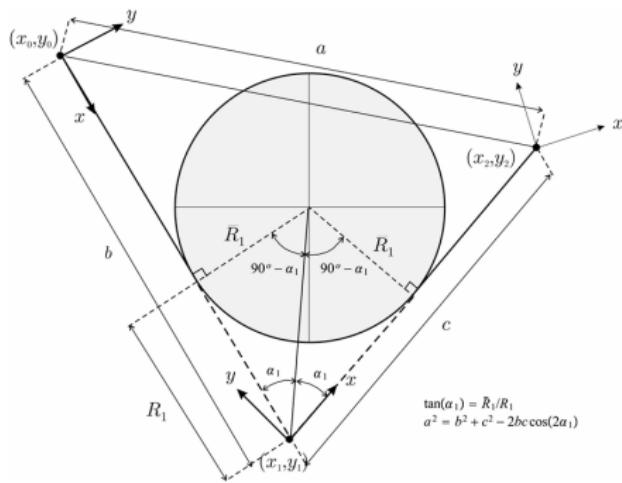


Guidance - Navigation - Control

Guidance - Path following

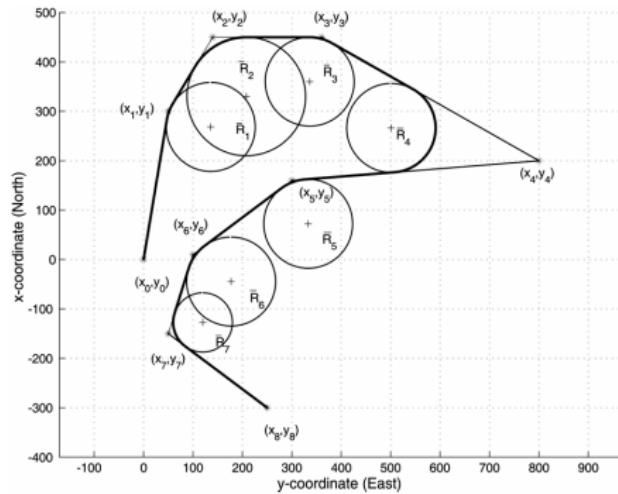
Path generation using straight lines and Circular Arcs

The shortest path (minimum time) between two configurations (x, y, ψ) of a craft moving at constant speed U is a path formed by straight lines and circular arc segments.



$$\tan(\alpha_1) = R_1/R_1$$

$$a^2 = b^2 + c^2 - 2bc \cos(2\alpha_1)$$

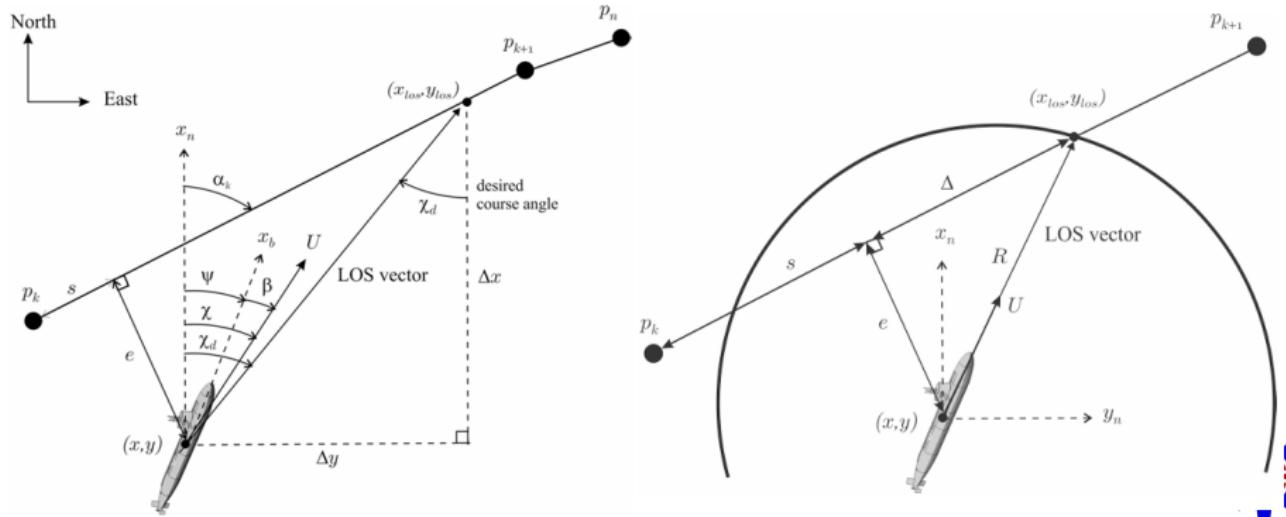


Guidance - Navigation - Control

Guidance - Path following

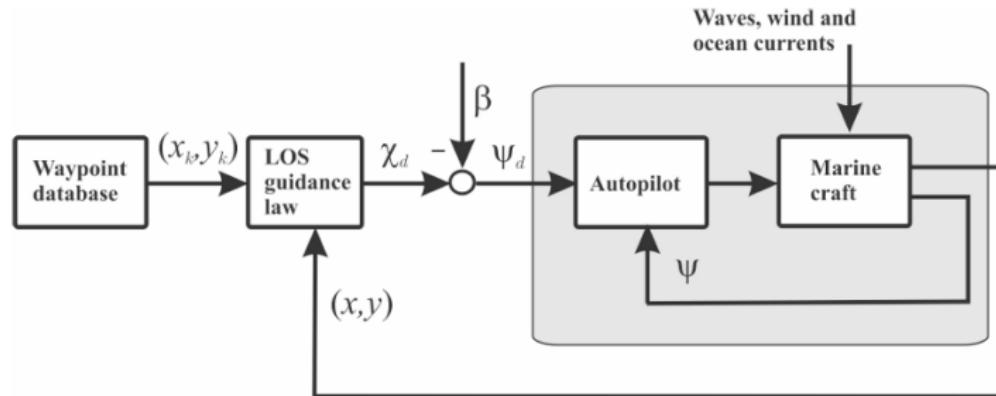
First of all, we consider a **Straight-line path** and introduce two different guidance principles, which could be used to steer along the LOS vector:

- Enclosure-based steering → find (x_{los}, y_{los}) base on a circle with radius $R > 0$ enclosing (x, y)
- Lookahead-based steering → steering law could be easy to modify



Guidance - Navigation - Control

Guidance - Path following



- Enclosure-based steering law $\rightarrow \psi_d = \text{atan}2(y_{los} - y, x_{los} - x)$
- Lookahead-based steering law $\rightarrow \psi_d = \chi_d - \beta = \alpha_k + \arctan(-\text{PID}(e)) - \beta$

Repeat again: β is the sideslip angle, which is calculated by $\beta = \arcsin(v/U)$



Guidance - Navigation - Control

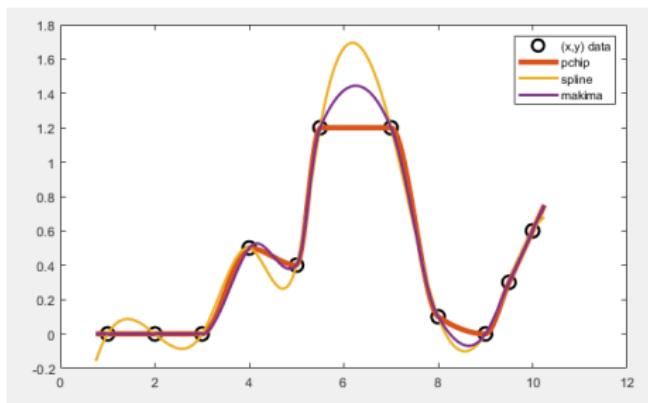
Guidance - Path following

In the following, **Path-Following for Curved Paths** is introduced.

MATLAB provide different methods for interpolation. For example, three methods for path generation are the cubic spline interpolant (`spline.m`), piecewise cubic Hermite interpolating polynomial (`pchip.m`) and modified Akima piecewise cubic Hermite interpolation (`makima.m`)

```
x = [1 2 3 4 5 5.5 7 8 9 9.5 10];
y = [0 0 0 0.5 0.4 1.2 1.2 0.1 0 0.3 0.6];
xq = 0.75:0.05:10.25;
yqs = spline(x,y,xq);
yqp = pchip(x,y,xq);
yqm = makima(x,y,xq);

plot(x,y, 'ko', 'LineWidth', 2, 'MarkerSize', 10)
hold on
plot(xq,yqp, 'LineWidth', 4)
plot(xq,yqs,xq,yqm, 'LineWidth', 2)
legend('(x,y) data', 'pchip', 'spline', 'makima')
```



Guidance - Navigation - Control

Guidance - Path following

In other approach, which is called cubic polynomials, we consider a parametrized path as follows

$$x_d(\varpi) = \sum_{i=0}^3 a_i \varpi^i; \quad y_d(\varpi) = \sum_{i=0}^3 b_i \varpi^i \quad (98)$$

Then, the above constraints must be satisfied

$$x_d(\varpi_{k-1}) = x_{k-1}, \quad x_d(\varpi_k) = x_k, \quad (99)$$

$$\lim_{\varpi \rightarrow \varpi_k^-} \frac{dx_d(\varpi_k)}{d\varpi_k} = \lim_{\varpi \rightarrow \varpi_k^+} \frac{dx_d(\varpi_k)}{d\varpi_k} \quad (100)$$

$$\lim_{\varpi \rightarrow \varpi_k^-} \frac{d^2 x_d(\varpi_k)}{(d\varpi_k)^2} = \lim_{\varpi \rightarrow \varpi_k^+} \frac{d^2 x_d(\varpi_k)}{(d\varpi_k)^2} \quad (101)$$

$$\frac{dx_d(\varpi_k)}{d\varpi_k}(\varpi_0) = const, \quad \frac{dx_d(\varpi_k)}{d\varpi_k}(\varpi_n) = const \quad (102)$$

$$OR \quad \left(\frac{d^2 x_d(\varpi_k)}{(d\varpi_k)^2}(\varpi_0) = const, \quad \frac{d^2 x_d(\varpi_k)}{(d\varpi_k)^2}(\varpi_n) = const \right)$$



Guidance - Navigation - Control

Guidance - Path following

Figure 1

File Edit View Insert Tools Desktop Window Help

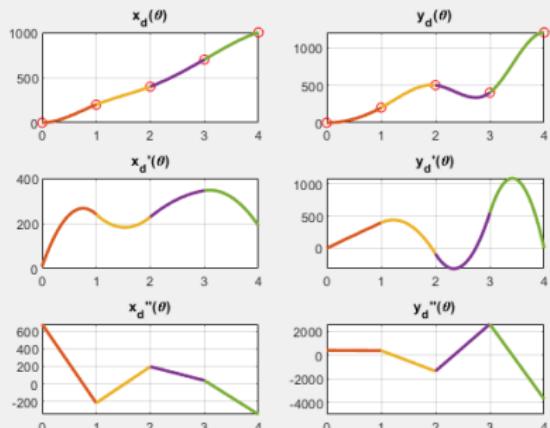
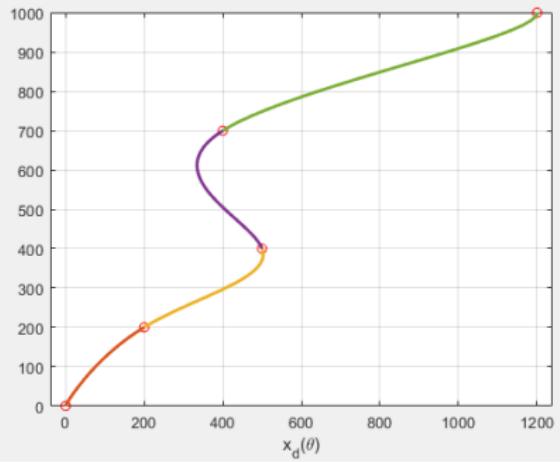


Figure 2

File Edit View Insert Tools Desktop Window Help



Transformation of Path to Reference Trajectories using Desired Speed Profiles

$$\dot{\varpi}(t) = \frac{U_d(t)}{\sqrt{\frac{dx_d(\varpi_k)}{d\varpi_k}(\varpi)^2 + \frac{dx_d(\varpi_k)}{d\varpi_k}(\varpi)^2}} \quad (104)$$

$$T\dot{U}_d(t) + U_d(t) = U_{ref} \quad (105)$$



In another view, the cubic polynomials approach could be formed by the **Nonlinear Constrained Optimization Problem**. Firstly, the constraint of the cubic polynomials approach could be represented by the the linear form

$$y = A(t_k, t_{k+1})x \quad (106)$$

where x is the vector of polynomial coefficients, and y , A are the derived constraints vector and matrix. Then, the optimization problem is formed

$$\text{minimize} \left[\left(A(t_k, t_{k+1})x - y \right)^\top \left(A(t_k, t_{k+1})x - y \right) \right] \quad (107)$$

with the constraint of velocities and accelerations.

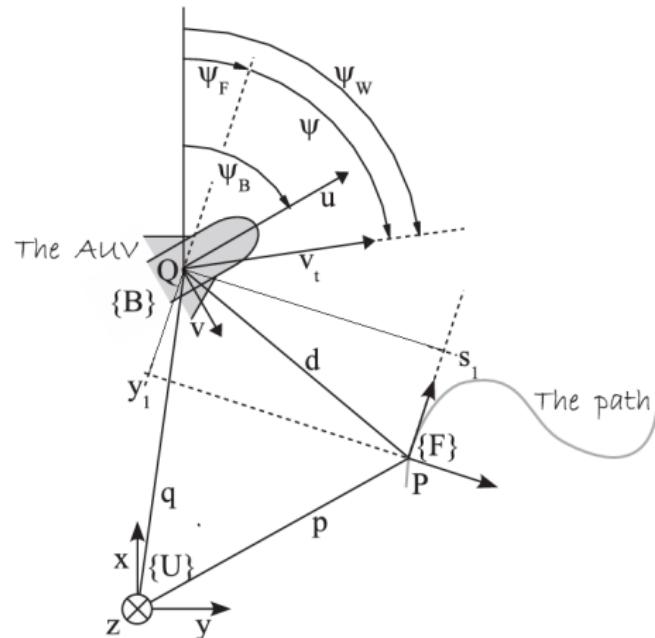


Guidance - Navigation - Control

Guidance - Path following

The solution to the problem of path-following proposed here builds on the following intuitive explanation a simple path-following controller should compute

- (i) the distance between the vehicle's center of mass Q and a point P , and
- (ii) the angle between the vehicle's total velocity vector v_t and the tangent to the path at P , and reduce both to zero. This motivates the development of the 'kinematic' model of the vehicle in terms of a Serret–Frenet frame $\{F\}$ that moves along the path; $\{F\}$ plays the role of the body axis of a 'virtual target vehicle' that should be tracked by the 'real vehicle'.



Guidance - Navigation - Control

Guidance - Path following

$$\dot{s}_1 = -\dot{s}(1 - c_c y_1) + v_t \cos \psi, \quad (108)$$

$$\dot{y}_1 = -c_c \dot{s}_1 + v_t \sin \psi, \quad (109)$$

$$\dot{\psi} = r + \dot{\beta} - c_c \dot{s}. \quad (110)$$

Assuming we have a precise estimation of the function $\mu(s)$, and given s , we compute

$$x_s(\mu) = \sum_{i=1}^n a_i \mu^i, \quad y_s(\mu) = \sum_{i=1}^n b_i \mu^i, \quad (111)$$

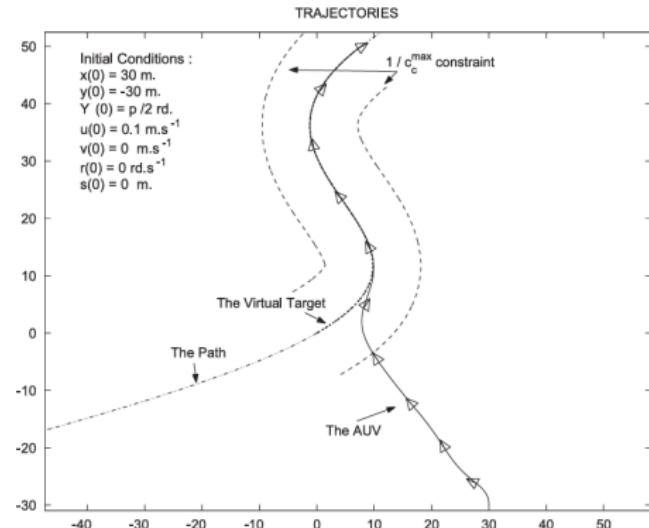
$$\psi_F(s) = \arctan \left(\frac{(y_s)'}{(x_s)'} \right), \quad (112)$$

$$c_c(s) = \frac{\partial \psi_F(s)}{\partial \mu} \frac{d\mu}{ds}, \quad (113)$$

$$\frac{\partial c_c(s)}{\partial s} = \frac{\partial c_c(s)}{\partial \mu} \frac{d\mu}{ds}, \quad (114)$$

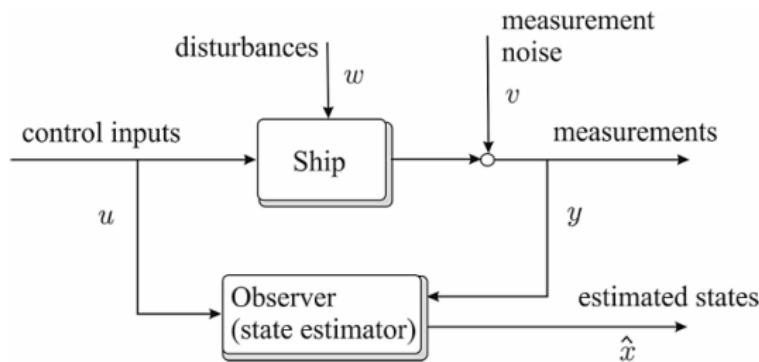
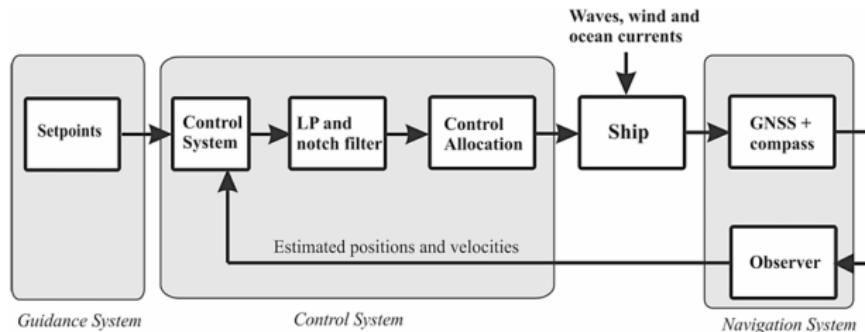
$$(x_s)' = \frac{dx_s}{d\mu}, \quad (y_s)' = \frac{dy_s}{d\mu} \quad (115)$$

$$\frac{d\mu}{ds} = \frac{1}{\sqrt{[(x_s)']^2 + [(y_s)']^2}}. \quad (116)$$



Guidance - Navigation - Control

Navigation



Low-Pass filtering: $\omega_b \ll \omega_e$,

$$\hat{y}(s) = h_{lp}(s)y(s) \quad (117)$$

where $h_{lp}(s)$ could be chosen as first-order low-pass filter, $h_{lp}(s) = \frac{1}{1 + T_f s}$ or the higher-order low-pass filter by using Butterworth filter, $h_{lf}(s) = \frac{1}{p(s)}$, with $p(s)p(-s) = 1 + (s/j\omega_f)^{2n}$

Cascade Low-Pass and Notch filtering:

$$\hat{y}(s) = h_{lp}(s)h_n(s)y(s) \quad (118)$$

with

$$h_n(s) = \frac{s^2 + 2\xi\omega_n s + \omega_n^2}{(s + \omega_n)^2}; \quad OR \quad h_n(s) = \prod_{i=1}^3 \frac{s^2 + 2\xi\omega_i s + \omega_i^2}{(s + \omega_i)^2} \quad (119)$$



Consider the linear time-invariant (LTI) system,

$$\dot{x} = Ax + Bu \quad (120)$$

$$y = Cx \quad (121)$$

and consider its dual system

$$\dot{\tilde{x}} = A^T \tilde{x} + C^T \tilde{u} \quad (122)$$

$$\tilde{y} = B^T \tilde{x} \quad (123)$$

Luenberger observer: could be designed as **pole placement control design** for the dual system.

Kalman observer: could be designed as **LQR control design** for the dual system.

Both of above-mentioned observers follow the below structure

$$\dot{z} = Az + Bu + L(y - Cz) \quad (124)$$

with L is observer gain.



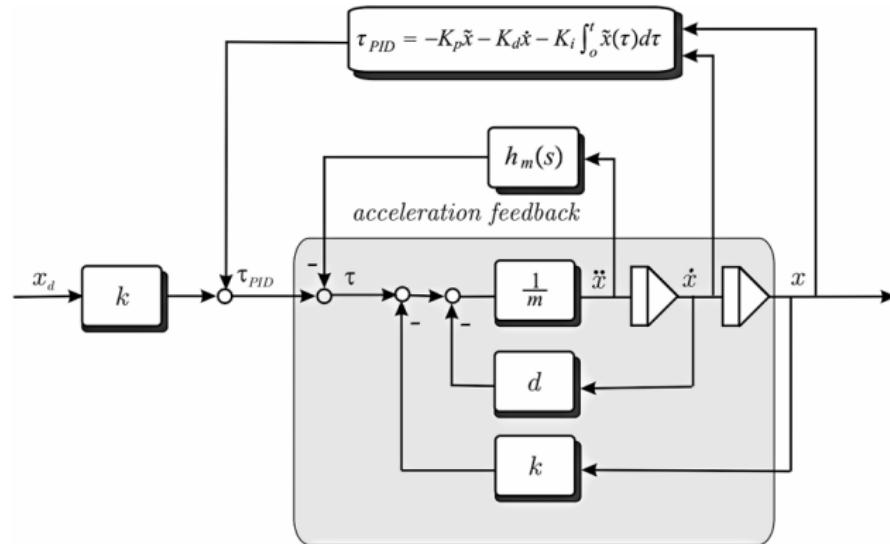
In other way, the advance observers that we could approach

- Extended Kalman Filter
- High-gain observer
- Sliding Mode Observer
- Extended State observer
- Fixed- and Finite-time observer
- ...



Guidance - Navigation - Control

Control - PID control



Closed-loop feedback system - PID control approach

$$\tau_{PID} = -K_p e - K_d \dot{e} - K_i \int_0^t e(\tau) d\tau \quad (125)$$



Guidance - Navigation - Control

Linear control approach

In the below section, we approach the control problem of linear system by Linear control technique.

First of all, consider the linearization algorithm for nonlinear system,

Assume that, the linear function at B is

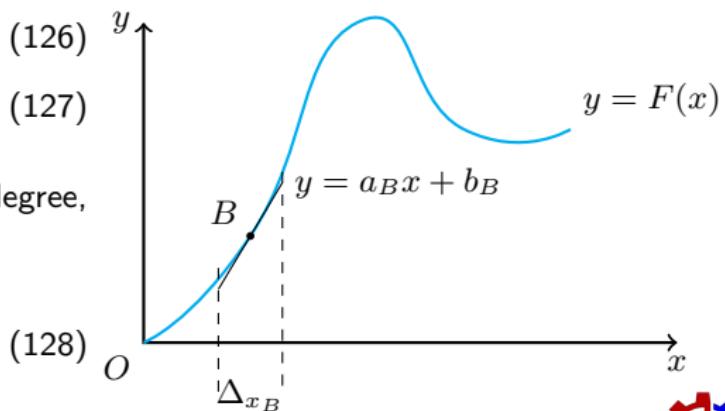
$$y = a_B x + b_B \quad (126)$$

$$\approx F(x) \Big|_{x \in \Delta x_B} \quad (127)$$

where a_B is the coefficient of first degree, easy to determine

$$a_B = \frac{\partial F(x)}{\partial x} \Big|_{x=x_B} \quad (128)$$

and b_B is the bias.



Guidance - Navigation - Control

Linear control approach

Consider a general system, which found in the above

$$\dot{x} = f(x) + g(x)\tau = F(x, \tau) \quad (129)$$

→ Let's approximate it to the linear original difference equation (ODE)

$$\dot{x} = Ax + B\tau \quad (130)$$

where, the state-space matrices are defined as

$$A = \left. \frac{\partial F(x, \tau)}{\partial x} \right|_{x=x_e, \tau=\tau_e} \quad (131)$$

$$B = \left. \frac{\partial F(x, \tau)}{\partial \tau} \right|_{x=x_e, \tau=\tau_e} \quad (132)$$



Guidance - Navigation - Control

State-Space-based Linear control approach

In the following step, we propose a linear control technique - pole placement control.

First, consider the state-space model, described as follows

$$\dot{x} = Ax + B\tau \quad (133)$$

As mention above, the control signal is

$$\tau = h(x) \quad (134)$$

such as, the system in (133) is stable.

For simplify the control signal, we assume that, the control law is a gain of state variables

$$\tau = h(x) = -Kx \quad (135)$$

This is the most simplify of control law!



Guidance - Navigation - Control

State-Space-based Linear control approach

With the control signal defined as $\tau = h(x) = -Kx$, the close-loop system becomes

$$\dot{x} = Ax - BKx \quad (136)$$

$$= (A - BK)x \quad (137)$$

Let $\tilde{A} = A - BK$. This ODE exist the solution as

$$x = e^{\tilde{A}t} = \sum_{i=0}^{\infty} \frac{\tilde{A}^i t^i}{i!} = P^{-1} \left(\sum_{i=0}^{\infty} \frac{L^i t^i}{i!} \right) P = P^{-1} e^{Lt} P \quad (138)$$

where

$$L = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_i) \quad (139)$$

λ_i are the eigenvalue of \tilde{A} with eigenvector P_i and

$$P = \begin{bmatrix} P_1^\top & P_2^\top & \dots & P_i^\top \end{bmatrix}^\top \quad (140)$$

Find K such as $\lim_{t \rightarrow \infty} x = 0$ that mean Find K such as $L < 0$



Similarly, by considering a optimization problem, Linear Quadratic Regulator is constructed.

First of all, consider linear time-invariant system

$$\frac{d}{dt}x = Ax + Bu \quad (141)$$

and control signal for this system

$$u = \omega - Gx \quad (142)$$

The LQR controller is a state feedback controller, with the control signal u found to be satisfied

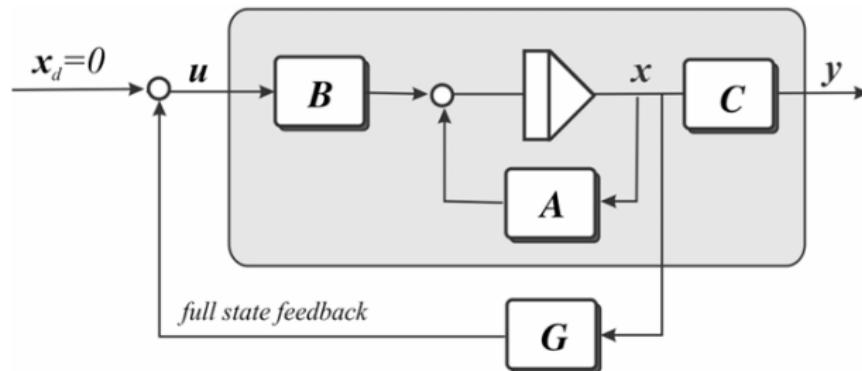
$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \rightarrow \min \quad (143)$$

J is the cost function, $Q = Q^T > 0$ and $R = R^T > 0$ are weight matrices.



Guidance - Navigation - Control

State-Space-based Linear control approach



The first step to find the LQR controller is to find the solution $P = P^T > 0$ of the Algebraic Riccati Equation (ARE)

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (144)$$

then, the controller matrix G can be found

$$G = R^{-1}B^T P$$



In the following section, a basic nonlinear control strategy that based on Sliding Mode Control is introduced.

For nonlinear control strategy, we consider the Sliding Mode Control technique. First of all, the control purpose is that the state variables track the desired states. That means if we consider the system, which is described by the above ODE

$$\ddot{x} = f(x, \dot{x}) + g(x, \dot{x})\tau \quad (146)$$

Assume that, the desired states is defined by x_d . Then, the error states are

$$e = x - x_d; \quad \dot{e} = \dot{x} - \dot{x}_d \quad (147)$$

Then, we define a sliding variable and a sliding surface with positive definite matrix $k > 0$ as follows

$$s = \dot{e} + ke; \quad \mathcal{S} = \{x, \dot{x} | s = 0\} \quad (148)$$

Notice that, by choosing a positive definite matrix k , this is easy to achieve that $s \rightarrow 0$, then $e \rightarrow 0$ and $\dot{e} \rightarrow 0$



In the following step, a control law is proposed to act the sliding variables to the sliding surface. First, we consider the Lyapunov candidate as follows

$$V = \frac{1}{2} s^\top s > 0 \quad (149)$$

And its derivative could be easy to achieve

$$\begin{aligned} \dot{V} &= s^\top \dot{s} = s^\top (\ddot{x} - \ddot{x}_d + k\dot{e}) \\ &= s^\top (f(x, \dot{x}) + g(x, \dot{x})\tau - \ddot{x}_d + k\dot{e}) \end{aligned} \quad (150)$$

Following Lyapunov stability theorem, let's choose the control signal as

$$\tau = g(x, \dot{x})^{-1} (-f(x, \dot{x}) + \ddot{x}_d - k\dot{e} - \eta s - \gamma \text{sign}(s)) \quad (151)$$

then, the derivative of Lyapunov candidate becomes

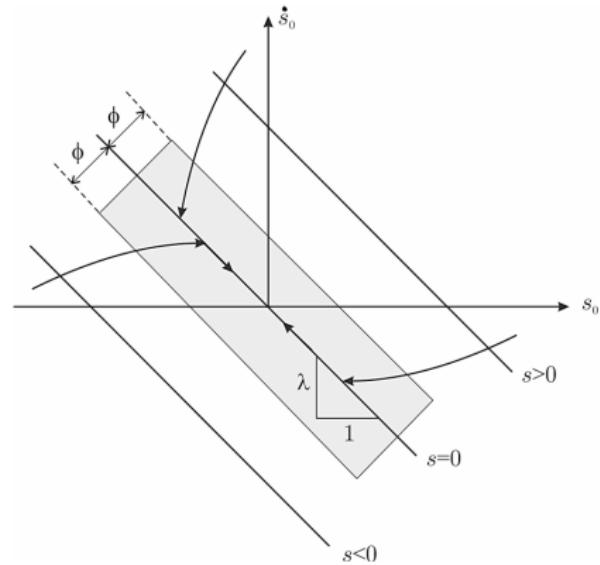
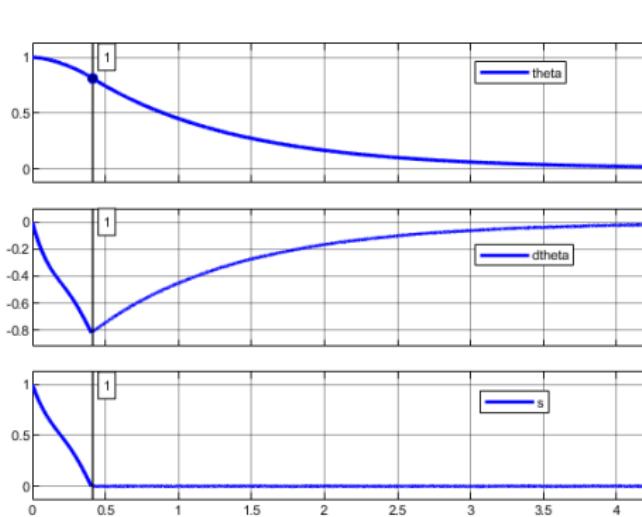
$$\dot{V} = s^\top (-\eta s - \gamma \text{sign}(s)) = -s^\top \eta s - \gamma |s| \quad (152)$$

→ if η and γ are positive definite, $\dot{V} < 0$, thus, $s \rightarrow 0$



Guidance - Navigation - Control

Nonlinear control approach



- Phase 1: The sliding variables reach the sliding surface, where the sliding variables are zeros
- Phase 2: The state errors reach to zeros such that $\dot{e} + ke = 0$



Guidance - Navigation - Control

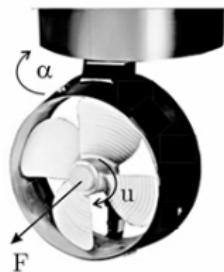
Forces and Moments allocation

Need a map from actuator to Force and Moment. That means

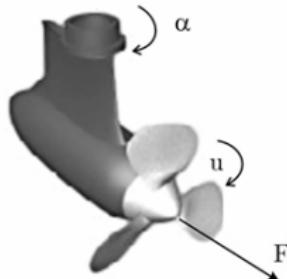
$$\text{Forces and Moments at center of mass: } \tau \in \mathbb{R}^6 \quad (153)$$

$$\text{Actuators: } u \in \mathbb{R}^n \quad (154)$$

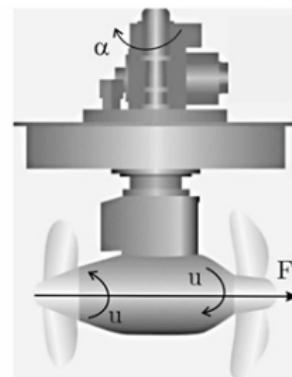
$$\text{Forces and Moments allocation: } f : \mathbb{R}^n \mapsto \mathbb{R}^6 \quad (155)$$



Azimuth thruster



Podded propeller



Contra-rotating propeller



Guidance - Navigation - Control

Forces and Moments allocation

Forces and Moment constraint: $\vec{F}_{\vec{k}} = \sum_i \vec{F}_i \vec{k}$ (156)

$$\vec{M}_{\vec{k}} = \sum_i M_i \vec{k} \quad (157)$$

$$M \text{ [Nm]} = F \text{ [N]} \times l \text{ [m]} \quad (158)$$

u_1, α_1	fore azimuth thruster	u_4	aft tunnel thruster
u_2	fore tunnel thruster	u_5	starboard main propeller
u_3, α_2	aft azimuth thruster	u_6	port main propeller

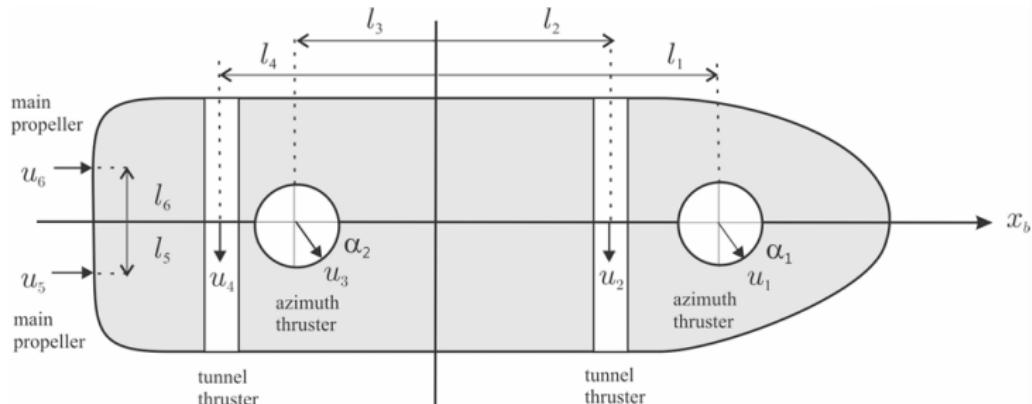


Table of Contents

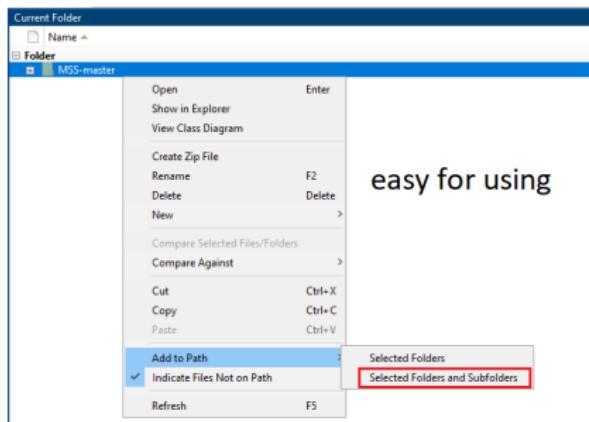
- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



Numerical simulations

MSS Toolbox

The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine control systems design. The m-files are compatible with the free software GNU Octave. MSS includes models for ships, underwater vehicles, uncrewed surface vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for time-domain simulation.



master		
File	Commit	Date
FDI	Update FDIradMod.m	9 months ago
GNC	Delete notchFilter.m	3 weeks ago
HYDRO	Delete Thumbs.db	last year
INS	Update SIMaidedINSeuler.m	last month
LIBRARY	Create highPassFilter.m	2 weeks ago
SIMULINK	Update Wave_init.m	5 months ago
VESSELS	Update SIMremus100.m	3 weeks ago
documentation	Delete README.txt	6 months ago
mssDemos	Update WaveDemo.m	5 months ago
mssExamples	Update exWaveMotionRAO.m	last month
.gitignore	Update .gitignore	10 months ago
Contents.m	Update Contents.m	8 months ago
How to install MSS for GNU Octave.md	Update How to install MSS for GNU Octave.md	6 months ago
How to install MSS for MATLAB.md	Update How to install MSS for MATLAB.md	6 months ago
LICENSE	Update LICENSE	last year
MSS Quick Reference.md	Update MSS Quick Reference.md	3 weeks ago
README.md	Update README.md	3 months ago
mssHelp.m	Create mssHelp.m	10 months ago
mssPath.m	Update mssPath.m	5 months ago

Numerical simulations

MSS Toolbox

The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine control systems design. The m-files are compatible with the free software GNU Octave. MSS includes models for ships, underwater vehicles, uncrewed surface vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for time-domain simulation.

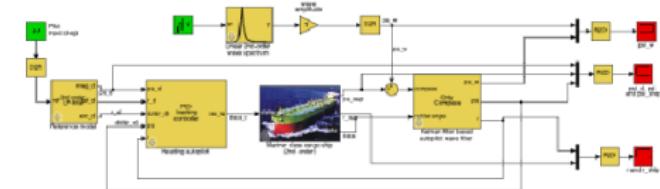
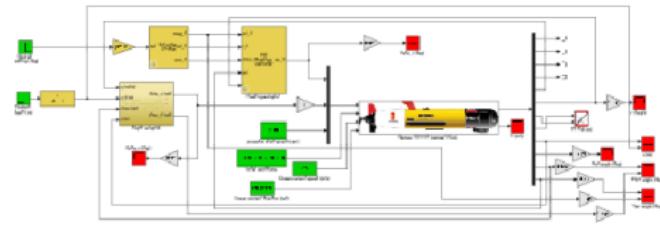
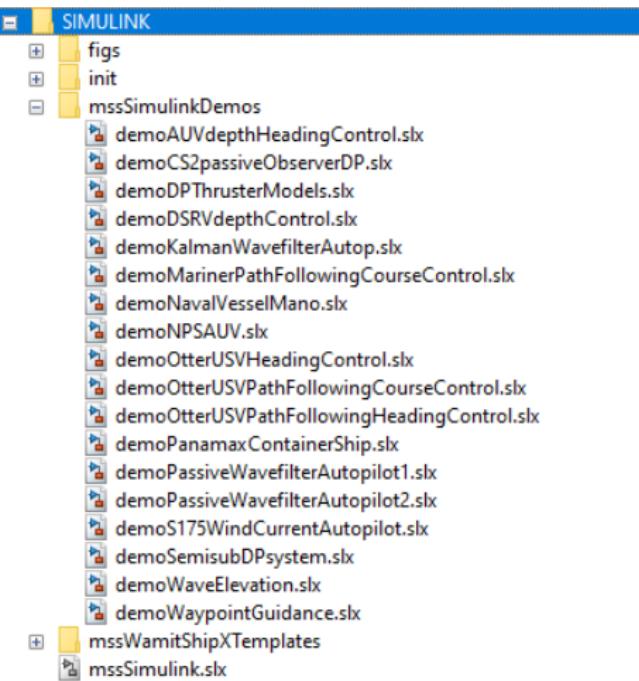
```
% Script for time-series simulations:  
%  
% SIMclarke83 - Simulate clarke83.m under PD control.  
% SIMdsrv - Simulate DSRV.m with an autopilot for depth control  
% using successive-loop closure.  
% SIMfrigate - Simulate frigate.m using a PID heading autopilot.  
% SIMmariner - Simulate mariner.m under PD control.  
% SIMotter - Simulate otter.m under feedback control.  
% SIMosv - Simulate osv.m under nonlinear DP control with  
% constrained control allocation (dynamic optimization).  
% SIMcontainer - Simulate container.m and Lcontainer.m under PD control.  
% SIMnavalvessel - Simulate navalvessel.m under PD control.  
% SIMnpsauv - Simulate npsauv.m with MIMO PID autopilots for depth  
and heading control, and 3-D straight-line path  
following using ALOS  
% SIMremus100 - Simulate remus100.m using autopilots for depth and  
heading control, and adaptive line-of-sight (ALOS)  
guidance laws for 3-D path-following control.  
% SIMrig - Simulate the 6-DOF semisubmersible model under PID  
control.  
% SIMsupply - Simulate the linear supply vessel model under DP control.  
% SIMtanker - Simulate the tanker model under DP control.
```

```
% The MSS marine craft models are compatibel with MATLAB and the free  
% software GNU Octave (www.octave.org):  
%  
% clarke83 - Linear maneuvering model parametrized using (L,B,T)  
found from linear regression of model tests (Clarke et  
al. 1983).  
% container - Nonlinear maneuvering model of a high-speed container  
ship, L = 175 m, including roll (Son and Nomoto 1982)  
% DSRV - Deep submergence rescue vehicle (DSRV), L = 5.0 m  
(Healey 1992).  
% frigate - Nonlinear autopilot model for a frigate, L = 100 m.  
% Lcontainer - Linearized model of a high-speed container ship,  
L = 175 m, including the roll mode (Son and Nomoto 1982).  
% mariner - Nonlinear maneuvering model for the Mariner class  
vessel, L = 160 m.  
% navalvessel - Nonlinear maneuvering model of a multipurpose naval  
vessel, L = 51.5 m.  
% npsauv - Naval Postgraduate School autonomous underwater vehicle  
(AUV), L = 5.3 m.  
% osv - Nonlinear model of an Offshore supply vessel (OSV),  
L = 83.0 m.  
% otter - OTTER small autonomous USV, L = 2.0 m.  
% remus100 - REMUS 100 autonomous underwater vehicle (AUV), L = 1.9 m  
% rig - Semisubmersible linear mass-damper-spring model,  
L = 84.6 m.  
% supply - Linear DP model of a supply vessel, L = 76.2 m.  
% tanker - Nonlinear course unstable maneuvering model of a  
tanker, L = 304 m.  
% zeefakkkel - Nonlinear autopilot model of a pleasure craft, L = 45 m.
```

Numerical simulations

MSS Toolbox

Simulink models are constructed by MATLAB R2024

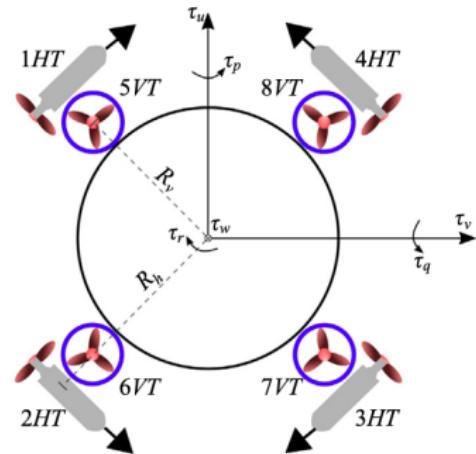


Numerical simulations

Example for ODIN

Forces distribution

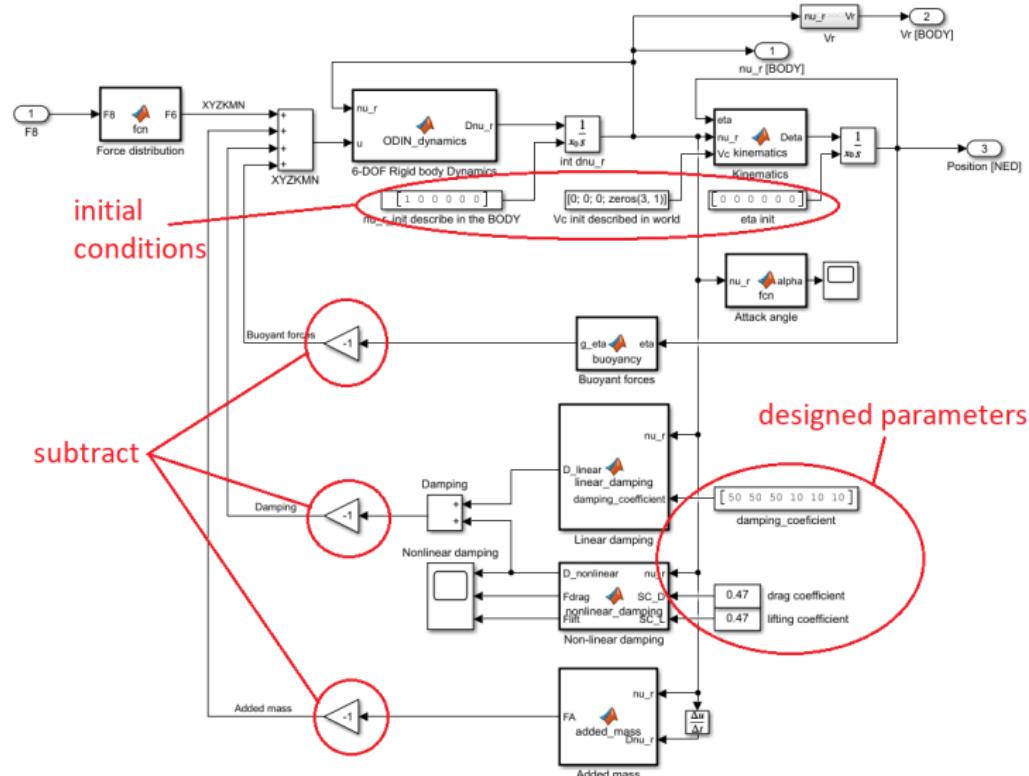
```
function F6 = fcn(F8)
s = 1/sqrt(2);
Rver = 0.63;
Rhorr = 0.89;
E = [s -s -s s 0 0 0 0;
      s s -s -s 0 0 0 0;
      0 0 0 0 -1 -1 -1 -1;
      0 0 0 0 Rver*s Rver*s -Rver*s -Rver*s;
      0 0 0 0 Rver*s -Rver*s -Rver*s Rver*s;
      Rhorr -Rhorr Rhorr -Rhorr 0 0 0 0];
F6 = E*F8;
```



Numerical simulations

Example for ODIN

Simulink simulation diagram



Numerical simulations

Example for ODIN

Dynamics and kinematics m-file of ODIN

```
function Dnu_r = ODIN_dynamics(nu_r, u)
%% Model parameters
rho = 1025;
g = 9.81;
R = 0.5;
V = 4/3*pi*R^3;
B = rho*g*V;
W = B;
m = W/g; neutral buoyant
%% Moment of inertia
I = (2/5) * m * R^2;
Ib = diag([I, I, I]);
rgb = [0; 0; 0];
%% Model dynamics
I3 = eye(3);
O3 = zeros(3);
nu1 = nu_r(1:3);
nu2 = nu_r(4:6);
skewMatrix = @(x) [0 -x(3) x(2); x(3) 0 -x(1); -x(2) x(1) 0];
%% Described in center of gravity (COM)
MRB_G = [m*I3 -m*skewMatrix(rgb); m*skewMatrix(rgb) Ib];
CRB_G = [m*skewMatrix(nu2), -m*skewMatrix(nu2)*skewMatrix(rgb);
skewMatrix(nu2)*skewMatrix(rgb), -skewMatrix(Ib*nu2)];
S = skewMatrix(rgb);
H = [eye(3) S';
zeros(3,3) eye(3) ];
%% Described in origin
MRB = H'*MRB_G*H;
CRB = H'*CRB_G*H;
%% M*nu + C*nu + G = u
Dnu_r = MRB^-1*(u - CRB*nu_r);
end
```

frame moving

```
function Data = kinematics(eta, nu_r, Vc)
I3 = eye(3);
O3 = zeros(3);
phi = eta(4); theta = eta(5); psi = eta(6);

% Rotation matrix about x-axis (Roll)
R_x = [1, 0, 0;
       0, cos(phi), -sin(phi);
       0, sin(phi), cos(phi)];
% Rotation matrix about y-axis (Pitch)
R_y = [cos(theta), 0, sin(theta);
       0, 1, 0;
       -sin(theta), 0, cos(theta)];
% Rotation matrix about z-axis (Yaw)
R_z = [cos(psi), -sin(psi), 0;
       sin(psi), cos(psi), 0;
       0, 0, 1];
% Combined rotation matrix (Z-Y-X convention)
R = R_z * R_y * R_x;
nu = nu_r + [R^(-1) O3; O3 I3]*Vc; NED frame
T = [1, sin(phi)*tan(theta), cos(phi)*tan(theta);
     0, cos(phi), -sin(phi);
     0, sin(phi)/cos(theta), cos(phi)/cos(theta)];
J = [R O3;
     O3 T];

Data = J*nu;
```



Numerical simulations

Example for ODIN

Buoyancy forces

```
function g_eta = buoyancy(eta)
phi = eta(4); theta = eta(5); psi = eta(6);
rho = 1025;
g = 9.81;
R = 0.5;
V = 4/3*pi*R^3;
B = rho*g*V;
W = B;
% m = W/g;
xg = 0;
yg = 0;
zg = 0;
xb = 0;
yb = 0;
zb = 0;
g_eta = [
    (W - B)*sin(theta);
    -(W - B)*cos(theta)*sin(phi);
    -(W - B)*cos(theta)*cos(phi);
    -(yg*W - yb*B)*cos(theta)*cos(phi) + (zg*W - zb*B)*cos(theta)*sin(phi);
    (zg*W - zb*B)*sin(theta) + (xg*W - xb*B)*cos(theta)*cos(phi);
    -(xg*W - xb*B)*cos(theta)*sin(phi) - (yg*W - yb*B)*sin(theta)];
end
```

position of CO and CB



Numerical simulations

Example for ODIN

Damping terms

```
function D_linear = linear_damping(nu_r, damping_coefficient)
Xu = damping_coefficient(1);
Yw = damping_coefficient(2);
Zw = damping_coefficient(3);
Kp = damping_coefficient(4);
Mq = damping_coefficient(5);
Nr = damping_coefficient(6);
```

design parameters

```
D = diag([Xu, Yw, Zw, Kp, Mq, Nr]);
D_linear = D*nu_r;
end
```

```
function [D_nonlinear, Fdrag, Flift] = nonlinear_damping(nu_r, SC_D, SC_L)
Vr = sqrt(nu_r(1:3)'*nu_r(1:3));
```

```
if nu_r(3) == 0 && nu_r(1) == 0
    alpha = 0;
else
    alpha = atan(nu_r(3)/nu_r(1));
end
```

attack angle

```
rho = 1025;
% SC_D = 0.47;
% SC_L = 0.47;
Fdrag = 1/2 * Vr^2 * rho * SC_D;
Flift = 1/2 * Vr^2 * rho * SC_L;
```

design parameters

```
D_nonlinear = [
    -Flift * sin(alpha) + Fdrag * cos(alpha);
    0;
    Flift * cos(alpha) + Fdrag * sin(alpha);
    zeros(3, 1)]; % 3x1 zero vector
end
```

Numerical simulations

Example for ODIN

Added mass

```
function FA = added_mass(nu_r, Dnu_r)

    ur = nu_r(4); vr = nu_r(5); wr = nu_r(6);

    rho = 1025;
    R = 0.5;

    I3 = eye(3); % 3x3 identity matrix
    O3 = zeros(3); % 3x3 zero matrix

    % Added mass matrix MA
    MA = rho * (2/3) * pi * R^3 * [I3, O3; O3, O3];

    % Coriolis term of added mass CA(nu_r)
    CA = rho * (2/3) * pi * R^3 * [
        0, 0, 0, 0, wr, -vr;
        0, 0, 0, -wr, 0, ur;
        0, 0, 0, vr, -ur, 0;
        0, -wr, vr, 0, 0, 0;
        wr, 0, -ur, 0, 0, 0;
        -vr, ur, 0, 0, 0, 0];

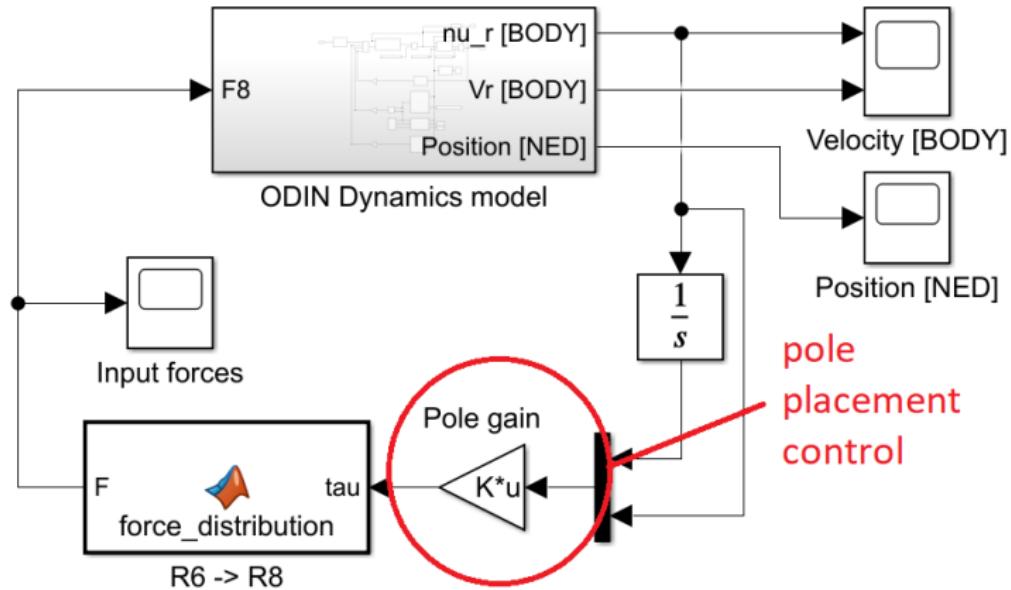
    FA = MA*Dnu_r + CA*nu_r;
end
```



Numerical simulations

Example for ODIN

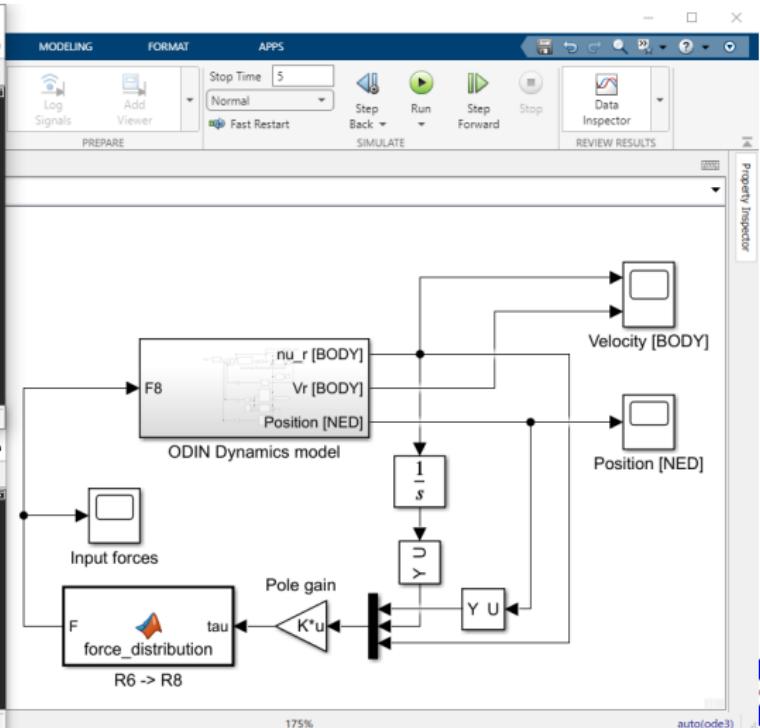
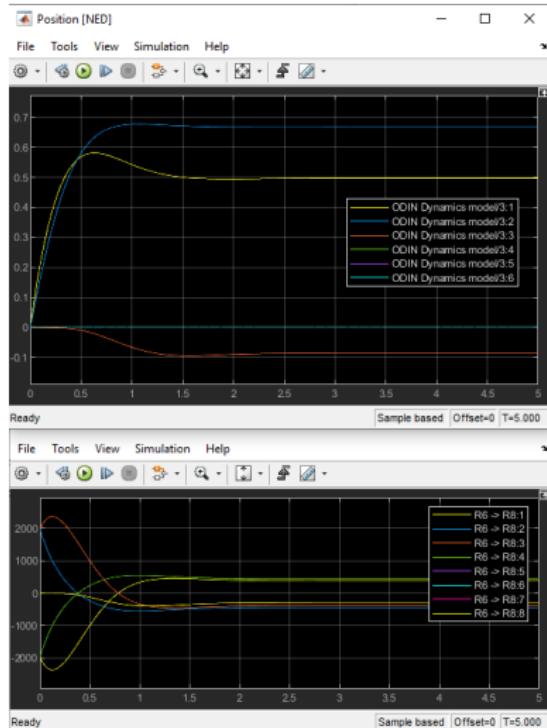
Simple control structure



Numerical simulations

Example for ODIN

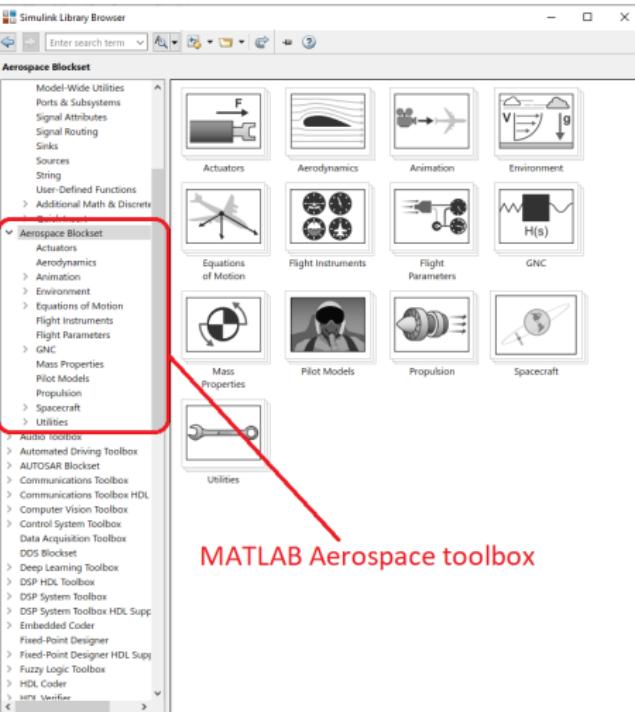
Simple control structure



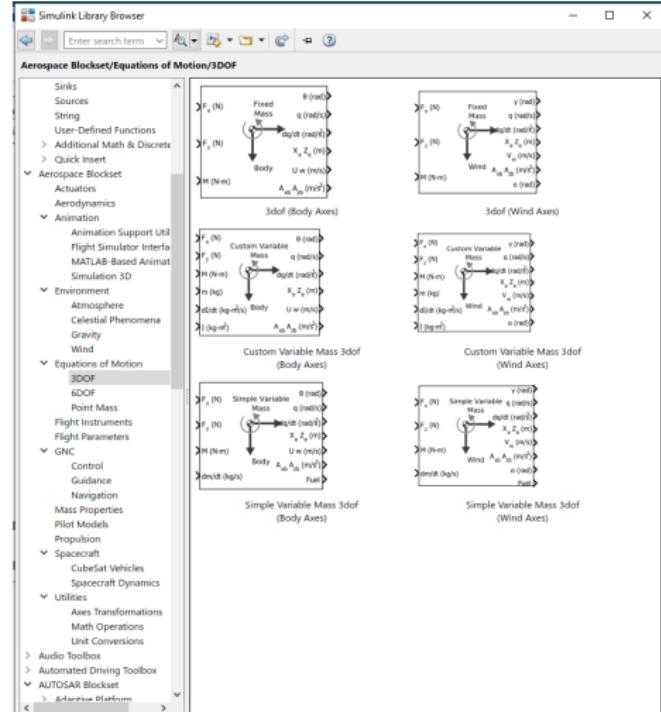
Numerical simulations

Aerospace toolbox

Aerospace Toolbox provides standards-based tools and functions for analyzing the motion, mission, and environment of aerospace vehicles.



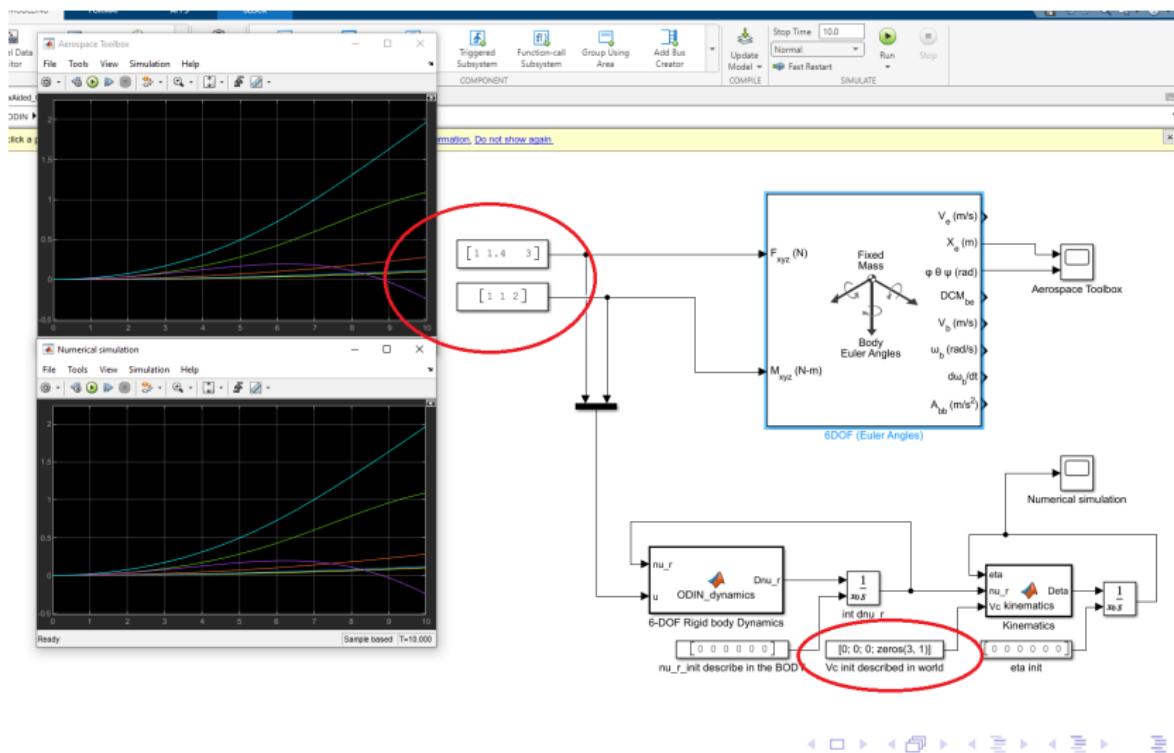
MATLAB Aerospace toolbox



Numerical simulations

Aerospace toolbox-aided simulation

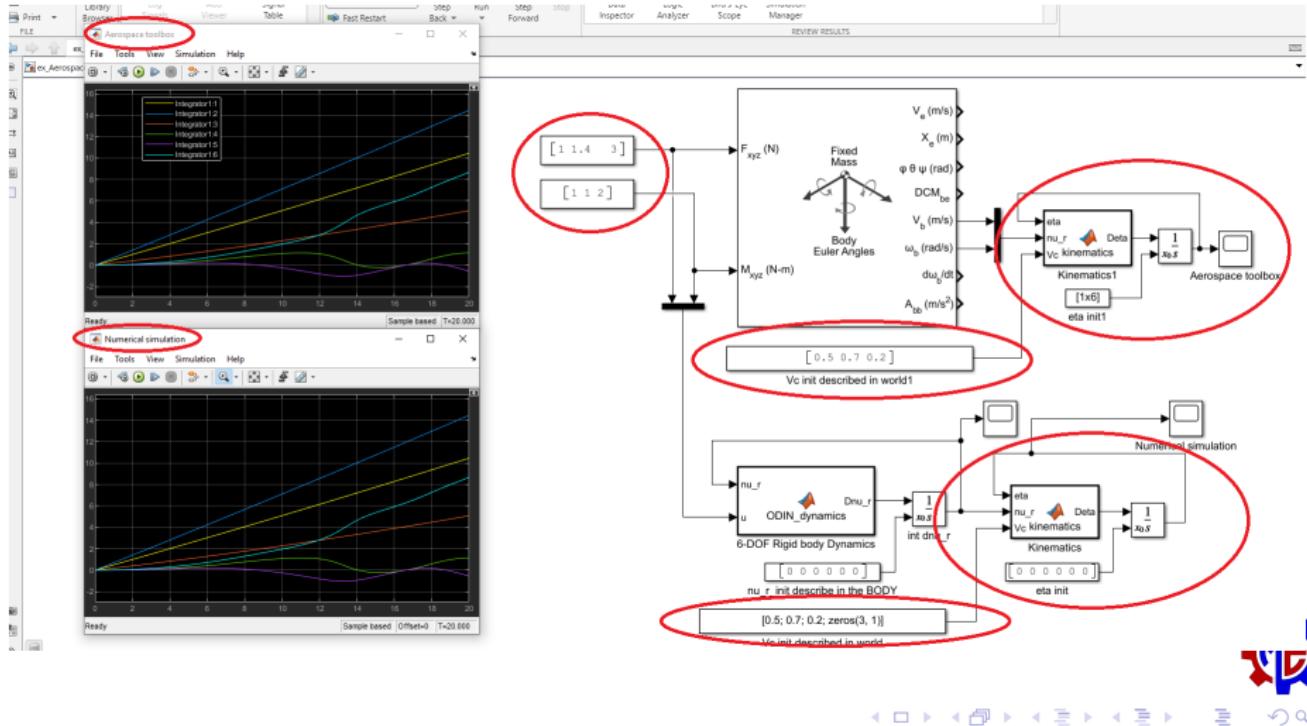
A comparison of Aerospace toolbox and traditional numerical simulation



Numerical simulations

Aerospace toolbox-aided simulation

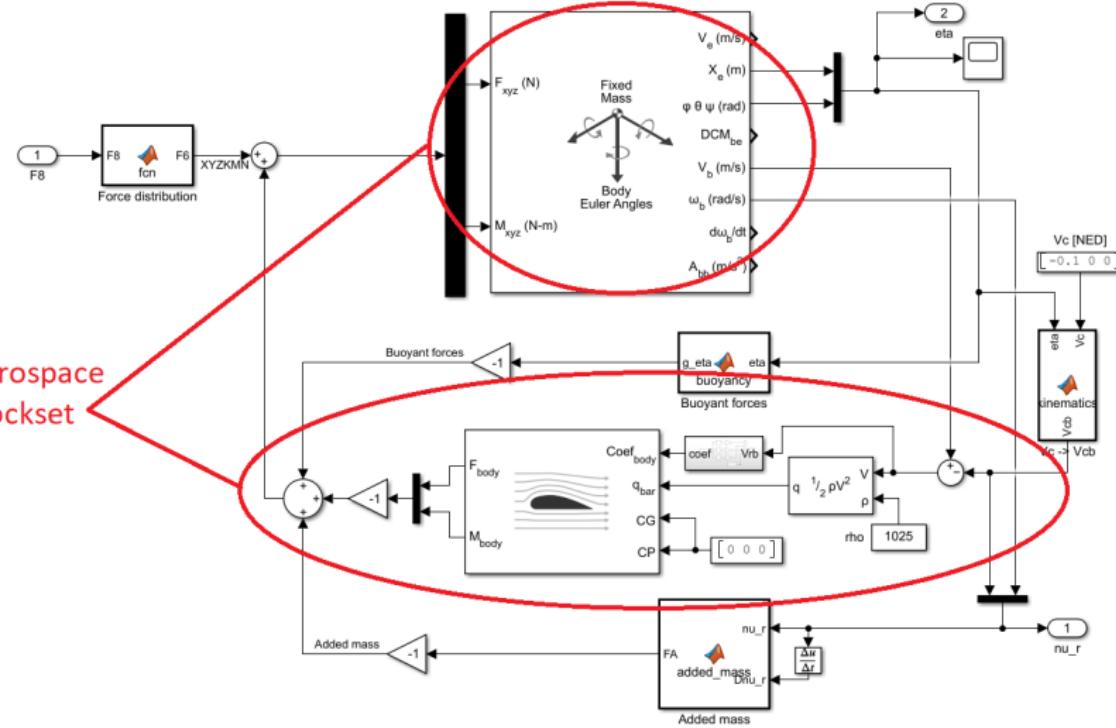
Ocean current consideration.



Numerical simulations

Aerospace toolbox-aided simulation

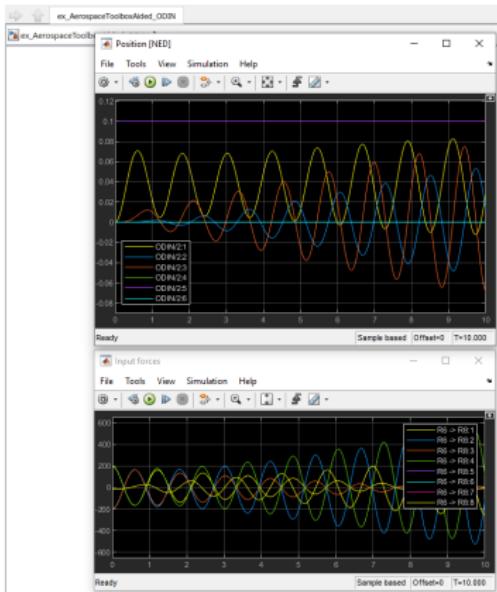
Modeling AUV with buoyant forces, nonlinear damping, and added mass.



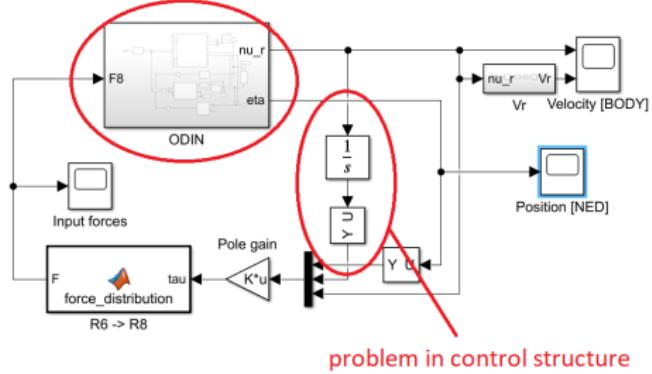
Numerical simulations

Aerospace toolbox-aided simulation

Simulation control structure



Aerospace toolbox-based model
(without linear damping)



problem in control structure



Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



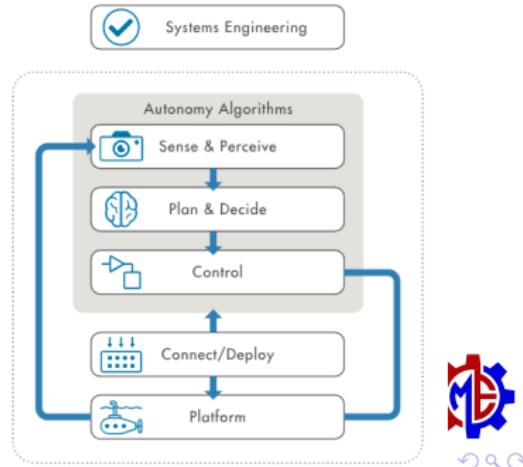
Quasi-Physical Simulation for UV motion control

Mathworks AUV



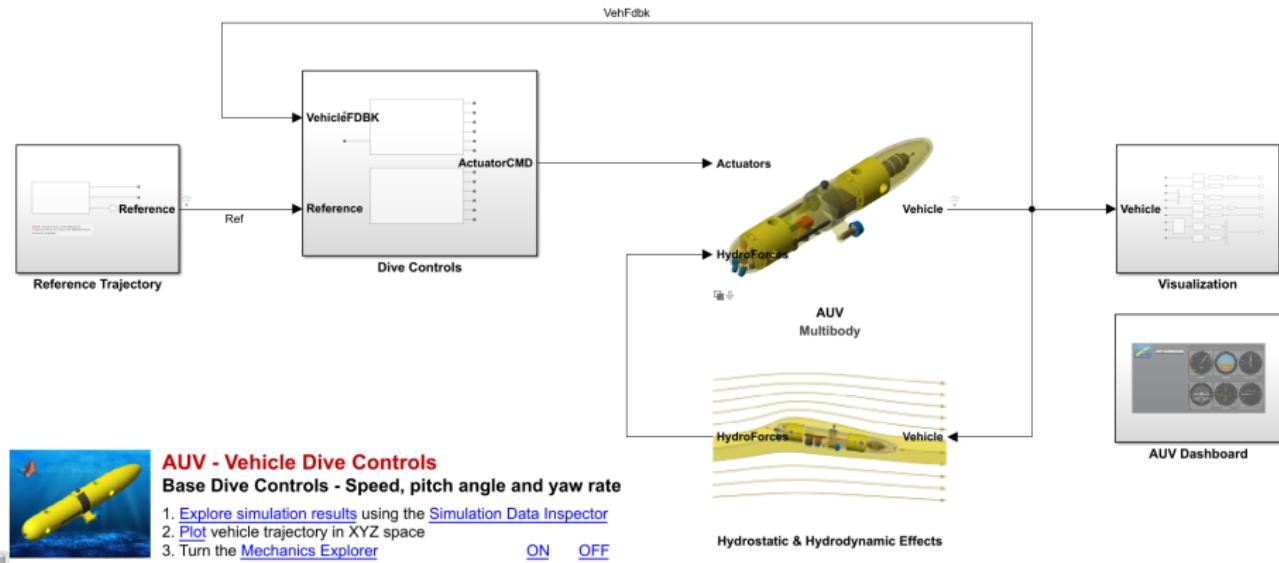
MATLAB and Simulink for Autonomous Underwater Vehicles

Interdisciplinary teams can use MATLAB and Simulink as a common integration environment throughout the entire autonomous underwater vehicle workflow. From systems engineering to platform modeling, environment simulation, and autonomy algorithm design, Model-Based Design helps you reduce risk and build confidence in system performance well in advance of the sea trial.



Quasi-Physical Simulation for UV motion control

Mathworks AUV



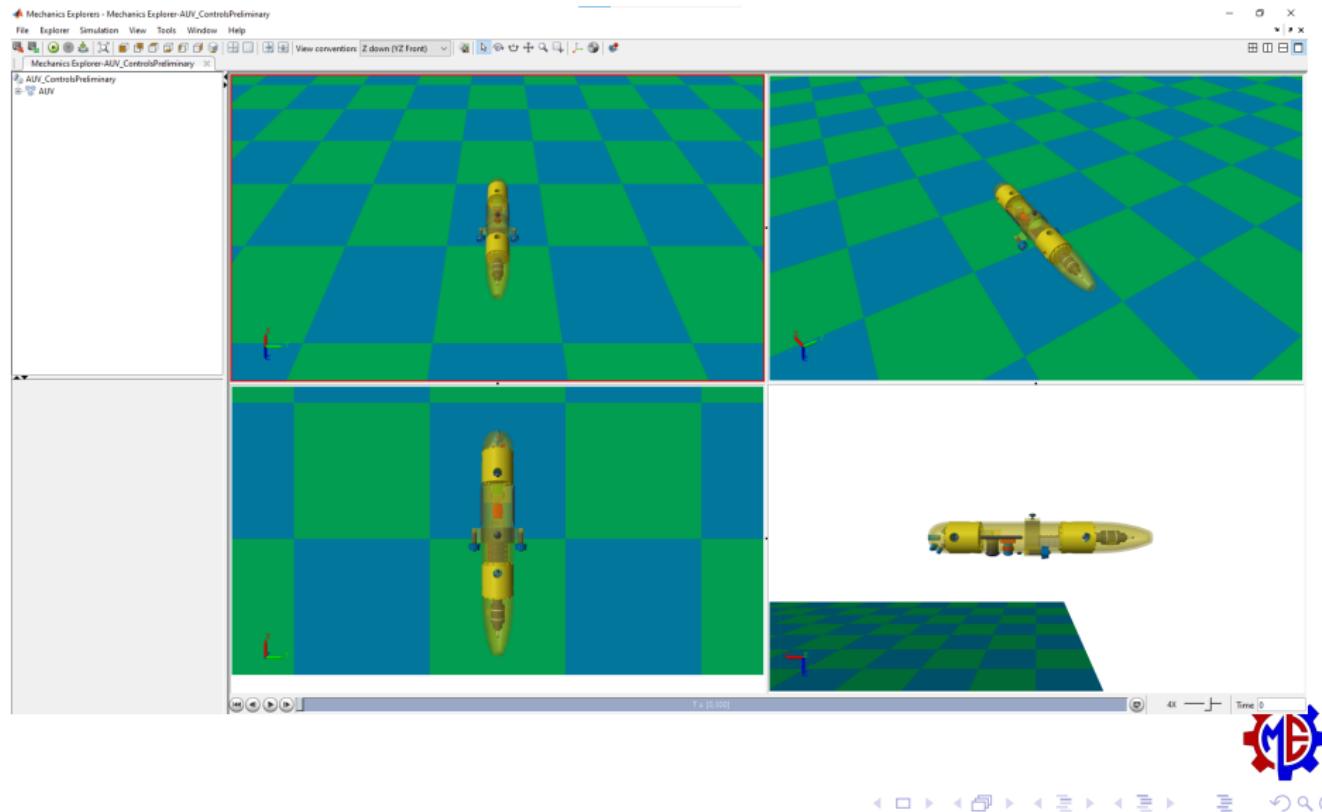
Simspace Multibody is used to simulate the mathematical model of the vehicle.

Simspace Multibody utilizes a combination of interconnected bodies (inertias), joints and constraints (relative DoFs) and force elements to solve the dynamics of the model



Quasi-Physical Simulation for UV motion control

Mathworks AUV



Quasi-Physical Simulation for UV motion control

Mathworks AUV

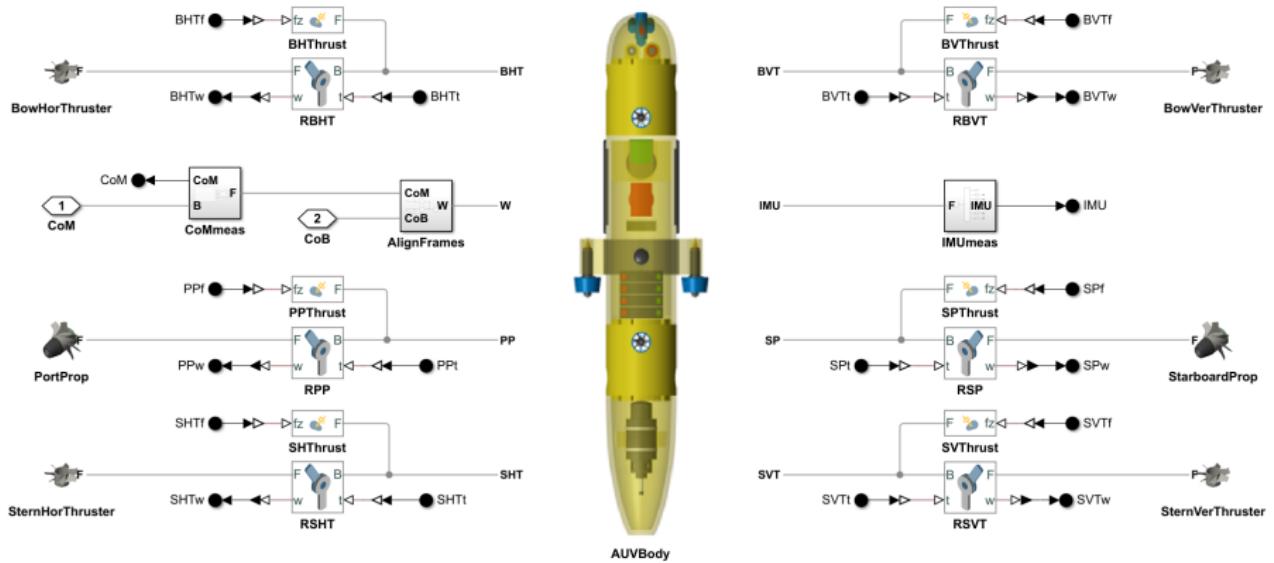
IMU located in the main hum, 6 actuators:

- BowVerThurster (vertical thurster located at the front of AUV)
- BowHorThurster (horizontal thurster located at the front of AUV)
- StarboardProp (Right Propeller)
- PortProp (Left Propeller)
- SternVerThurster (vertical thurster located at the back of AUV)
- SternHorThurster (horizontal thurster located at the back of AUV)



Quasi-Physical Simulation for UV motion control

Mathworks AUV



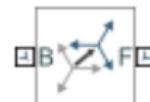
To describe the behavior of a multi-body system, we define the corresponding coordinates and attached forces



Quasi-Physical Simulation for UV motion control

Mathworks AUV

- Body frames can be created using body elements block with preset shapes and connected using rigid transform blocks



- Joint blocks connect the frames and impose kinetic constraints and determine how they move relative to each other



- The External Force and Torque represents forces or torques applied on the multibody model.



- File Solid Block associates individual part files from CAD formats to body blocks in the model.



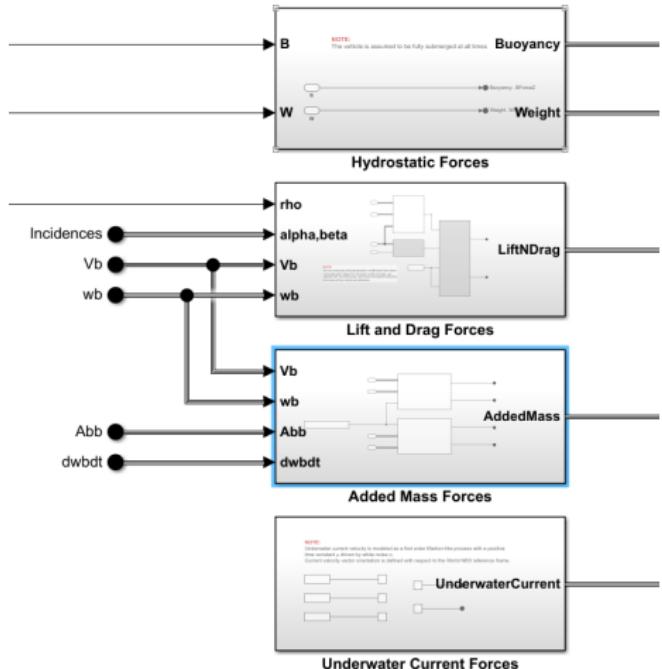
Quasi-Physical Simulation for UV motion control

Mathworks AUV

Represents external forces of the marine environment.

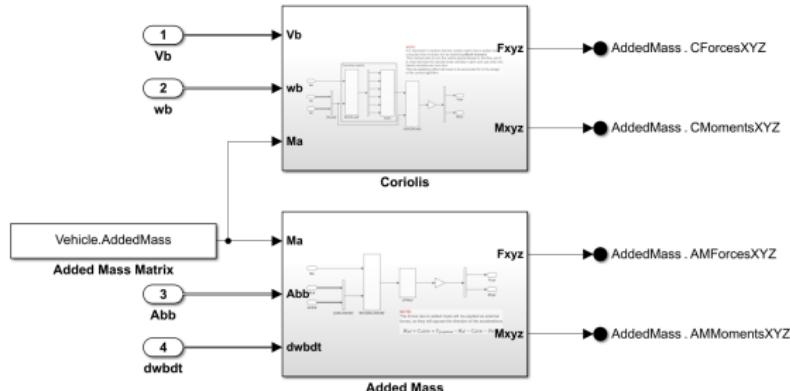
Forces consist of:

- Gravity and buoyancy effects
- Lift and drag forces
- Added mass effects
- Underwater current forces



Quasi-Physical Simulation for UV motion control

Mathworks AUV



Mathematical model

$$(M_{RB} + M_A)\dot{\nu} + (C_{RB} + C_A)(\nu)\nu + D(\nu_r)\nu_r + \underbrace{g(\eta) + g_o}_{\text{hydrostatic forces}} = F_{propulsion} \quad (159)$$

where M_A is the added mass matrix and C_A is the coriolis effect induced by the added mass



Quasi-Physical Simulation for UV motion control

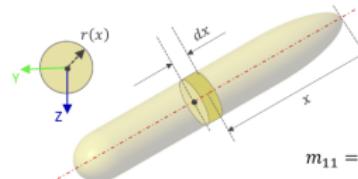
Mathworks AUV

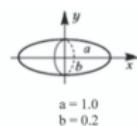
Added mass

The elongated myring shape of the hull results in

$$M_A = \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 & 0 & m_{26} \\ 0 & 0 & m_{33} & 0 & m_{35} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{53} & 0 & m_{55} & 0 \\ 0 & m_{62} & 0 & 0 & 0 & m_{66} \end{bmatrix}$$

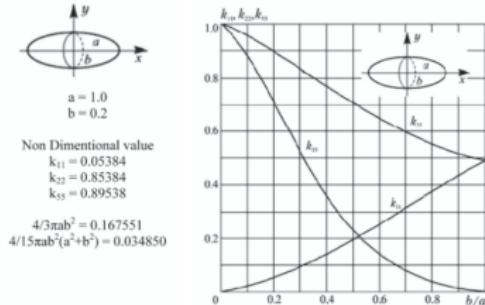
$$m_{22} = m_{53} \text{ and } m_{55} = m_{66}$$


$$m_{11} = k_{11} \frac{4}{3} \pi \rho r_1 r_z^2$$
$$m_{22} = \int_0^L \pi \rho r(x)^2 dx \quad m_{26} = \int_0^L x \pi \rho r(x)^2 dx$$
$$m_{66} = \int_0^L x^2 \pi \rho r(x)^2 dx \quad m_{35} = - \int_0^L x \pi \rho r(x)^2 dx$$



Non Dimensional value
 $k_{11} = 0.05384$
 $k_{22} = 0.85384$
 $k_{35} = 0.89538$

$$\frac{4}{3}\pi ab^2 = 0.167551$$
$$\frac{4}{15}\pi ab^2(a^2+b^2) = 0.034850$$



Quasi-Physical Simulation for UV motion control

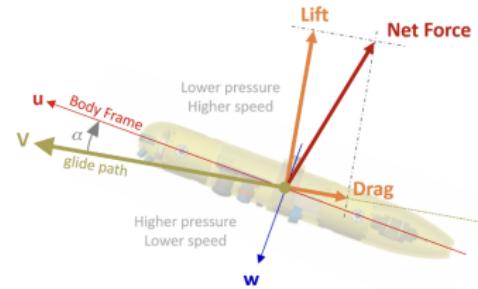
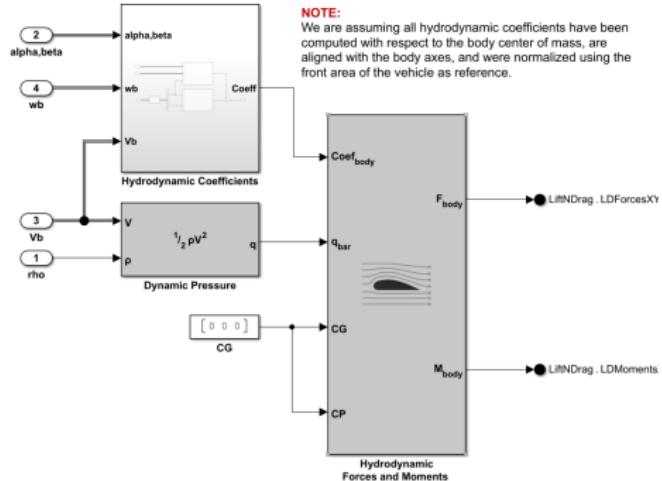
Mathworks AUV

Drag and lift forces

$$F_{drag} = \frac{1}{2} \rho V^2 C_d A$$

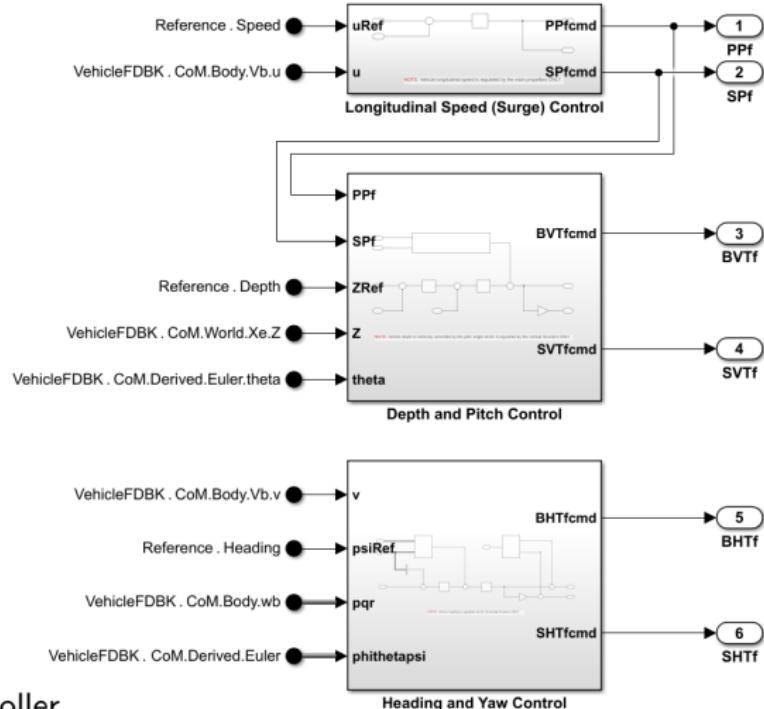
$$F_{lift} = \frac{1}{2} \rho V^2 C_l A$$

where C_d and C_l are normally from experiments and used in practice



Quasi-Physical Simulation for UV motion control

Mathworks AUV



3 Controllers

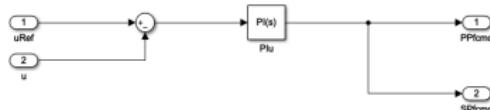
- Surge Controller
- Heave and Pitch Controller
- Yaw Controller



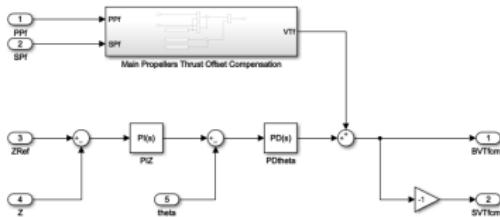
Quasi-Physical Simulation for UV motion control

Mathworks AUV

- Surge Controller controls both propellers



- Heave and Pitch Controller controls vertical thrusters



- Yaw Controller controls horizontal thrusters

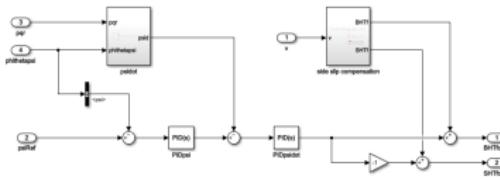


Table of Contents

- 1 Introduction
- 2 Fundamental knowledge
- 3 6-DOF model of a Rigid Body
- 4 Ocean dynamics
- 5 Environmental Disturbances
- 6 An example of the ODIN
- 7 Guidance - Navigation - Control
- 8 Numerical simulations
- 9 Quasi-Physical Simulation for UV motion control
- 10 An example for GNC-integrated Quasi-Physical simulation



An example for GNC-integrated Quasi-Physical simulation

[Coming soon!!]



Thanks for your attention! Any questions?

Hope you slept comfortably!

